(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2001 (13.09.2001)

PCT

(10) International Publication Number
WO 01/67186 A1

(51) International Patent Classification[7]: G05B 13/02, G06F 17/30

(21) International Application Number: PCT/IT00/00078

(22) International Filing Date: 9 March 2000 (09.03.2000)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicants *(for all designated States except US)*: **STMICROELECTRONICS S.R.L.** [IT/IT]; Via C. Olivetti, 2, I-20041 Agrate Brianza (IT). **YAMAHA MOTOR EUROPE N.V.** [NL/NL]; Koolhovenlaan 101, NL-1119 NC Schiphol-Rijk (NL).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **ULYANOV, Serguei** [RU/IT]; Via Matteotti, 56, I-26013 Crema (IT). **RIZZOTTO, Gianguido** [IT/IT]; Via Belvedere, 22/24, I-23862 Civate (IT). **KURAWAKI, Ichiro** [JP/IT]; Viale S. Gimignano, 15/A, I-20146 Milano (IT). **PANFILOV, Serguei** [RU/IT]; Via Piacenza, 37/C, I-26013 Crema (IT). **GHISI, Fabio** [IT/IT]; Via Giovanni XXIII, 45, I-25020 Flero (IT). **AMATO, Paolo** [IT/IT]; Via Roma, 26/A, I-20051 Limbiate (IT). **PORTO, Massimo** [IT/IT]; Viale Tirreno, 3, I-95123 Catania (IT).

(74) Agents: **PELLEGRI, Alberto** et al.; Societa' Italiana Brevetti S.p.A., Piazza Repubblica, 5, I-21100 Varese (IT).

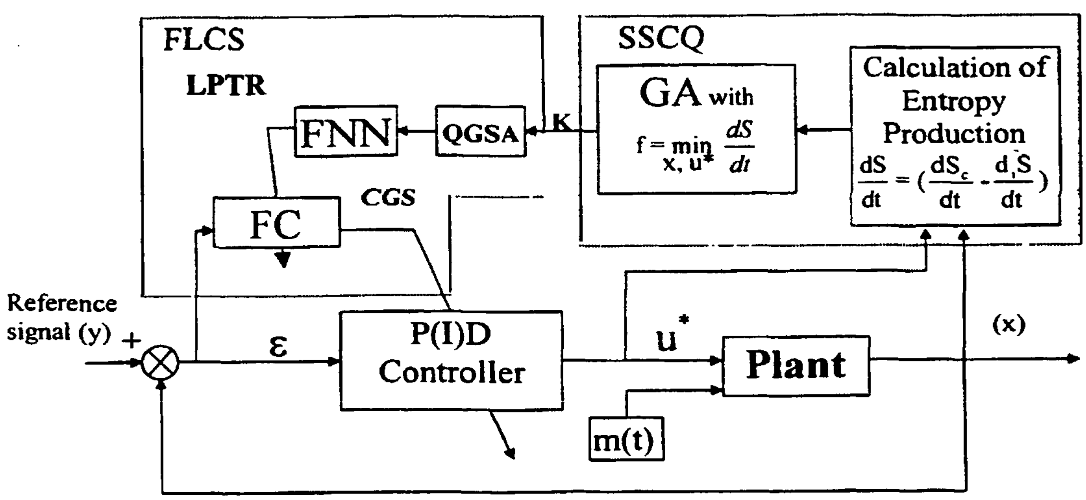
(81) Designated States *(national)*: CN, JP, US.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

**Published:**
— *with international search report*

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

# (54) Title: METHOD AND HARDWARE ARCHITECTURE FOR CONTROLLING A PROCESS OR FOR PROCESSING DATA BASED ON QUANTUM SOFT COMPUTING




(57) Abstract: A method of controlling a process (Plant) driven by a control signal (U*) for producing a corresponding output comprises producing an error signal (ε) as a function of a state of the process (X) and of a reference signal (Y), generating a control signal (U*) as a function of the error signal (ε) and of a parameter adjustment signal (CGS) and feeding it to the process (Plant), deriving a signal (S) representative of a quantity to be minimized calculated by processing paired values of the state (X) of the process and the control signal (U*), calculating a correction signal (K2) from a set of several different values of the control signal (U*), that minimize the derived signal (S) to be minimized, calculating the parameter adjustment signal (CGS) by a neural network and fuzzy logic processor from the error signal (ε) and correction signal (K2). The method of the invention is characterized in that the correction signal (K2) is periodically calculated by a Quantum Genetic Search Algorithm consisting in a merging of a genetic algorithm and a quantum search algorithm. A hardware embodiment of the method of the invention has been disclosed.

WO 01/67186 A1

# "METHOD AND HARDWARE ARCHITECTURE FOR CONTROLLING A PROCESS OR FOR PROCESSING DATA BASED ON QUANTUM SOFT COMPUTING"

## FIELD OF THE INVENTION

This invention generally relates to a method and a hardware for controlling a process or for processing data in a database, and more specifically to control a process and/or including search-of-minima intelligent operations.

The method of the invention is outstandingly useful for minimum searching among a set of values and in particular for realizing hardware control systems exploiting artificial intelligence to robust smart control a nonlinear process and/or to search in a database.

## BACKGROUND OF THE INVENTION

Feedback control systems are widely used to maintain the output of a dynamic system at a desired value in spite of external disturbances that would displace it from the desired value. For example, a household space heating furnace, controlled by a thermostat, is an example of a feedback control system. The thermostat continuously measures the air temperature inside the house, and when the temperature falls below a desired minimum temperature the thermostat turns the furnace on. When the interior temperature reaches the desired minimum temperature, the thermostat turns the furnace off. The thermostat-furnace system maintains the household temperature at a substantially constant value in spite of external disturbances such as a drop in the outside temperature. Similar types of feedback controls are used in many applications.

A central component in a feedback control system is a controlled object, a machine or a process that can be defined as a "plant", whose output variable is to be controlled. In the above example, the "plant" is the house, the output variable is the interior air temperature in the house and the disturbance is the flow of heat

(dispersion) through the walls of the house. The plant is controlled by a control system. In the above example, the control system is the thermostat in combination with the furnace. The thermostat-furnace system uses simple on-off feedback control system to maintain the temperature of the house. In many control environments, such as motor shaft position or motor speed control systems, simple on-off feedback control is insufficient. More advanced control systems rely on combinations of proportional feedback control, integral feedback control, and derivative feedback control. A feedback control based on a sum of proportional, plus integral, plus derivative feedback, is often referred as a PID control.

A PID control system is a linear control system that is based on a dynamic model of the plant. In classical control systems, a linear dynamic model is obtained in the form of dynamic equations, usually ordinary differential equations. The plant is assumed to be relatively linear, time invariant, and stable. However, many real-world plants are time varying, highly nonlinear, and unstable. For example, the dynamic model may contain parameters (e.g., masses, inductance, aerodynamics coefficients, etc.) which are either only approximately known or depend on a changing environment. If the parameter variation is small and the dynamic model is stable, then the PID controller may be satisfactory. However, if the parameter variation is large or if the dynamic model is unstable, then it is common to add adaptive or intelligent (AI) control functions to the PID control system.

AI control systems use an optimizer, typically a nonlinear optimizer, to program the operation of the PID controller and thereby improve the overall operation of the control system.

Classical advanced control theory is based on the assumption that near equilibrium points all controlled “plants” can be approximated as linear systems. Unfortunately, this assumption is rarely true in the real world. Most plants are highly nonlinear, and often do not have simple control algorithms. In order to meet these needs for a nonlinear control, systems have been developed that use

soft computing concepts such as genetic algorithms (GA), fuzzy neural networks (FNN), fuzzy controllers and the like. By these techniques, the control system evolves (changes) in time to adapt itself to changes that may occur in the controlled "plant" and/or in the operating environment.

A control system for controlling a plant based on soft computing is depicted in Fig. 1.

Using a set of inputs, and a fitness function, the genetic algorithm works in a manner similar to an evolutional process to arrive at a solution which is, hopefully, optimal.

The genetic algorithm generates sets of "chromosomes" (that is, possible solutions) and then sorts the chromosomes by evaluating each solution using the fitness function. The fitness function determines where each solution ranks on a fitness scale. Chromosomes (solutions) which are more fit, are those which correspond to solutions that rate high on the fitness scale. Chromosomes which are less fit, are those which correspond to solutions that rate low on the fitness scale.

Chromosomes that are more fit are kept (survive) and chromosomes that are less fit are discarded (die). New chromosomes are created to replace the discarded chromosomes. The new chromosomes are created by crossing pieces of existing chromosomes and by introducing mutations.

The PID controller has a linear transfer function and thus is based upon a linearized equation of motion for the controlled "plant". Prior art genetic algorithms used to program PID controllers typically use simple fitness and thus do not solve the problem of poor controllability typically seen in linearization models. As is the case with most optimizers, the success or failure of the optimization often ultimately depends on the selection of the performance (fitness) function.

Evaluating the motion characteristics of a nonlinear plant is often difficult, in part due to the lack of a general analysis method. Conventionally, when controlling a plant with nonlinear motion characteristics, it is common to find certain equilibrium points of the plant and the motion characteristics of the plant are linearized in a vicinity near an equilibrium point. Control is then based on evaluating the pseudo (linearized) motion characteristics near the equilibrium point. This technique is scarcely, if at all, effective for plants described by models that are unstable or dissipative.

Computation of optimal control based on soft computing includes GA as the first step of global search for optimal solution on a fixed space of positive solutions. The GA searches for a set of control weights for the plant. Firstly the weight vector $K = \{k_1, \ldots, k_n\}$ is used by a conventional proportional-integral-differential (PID) in the generation of a signal $\delta(K)$ which is applied to the plant. The entropy $S(\delta(K))$ associated to the behaviour of the plant on this signal is assumed as a fitness function to minimize. The GA is repeated several times at regular time intervals in order to produce a set of weight vectors. The vectors generated by GA are then provided to a FNN and the output of the FNN to a fuzzy controller. The output of the fuzzy controller is a collection of gain schedules for the *PID* - controller that controls the plant. For soft computing systems based on *GA*, there is very often no real control law in the classical control sense, but rather, control is based on a physical control law such as minimum entropy production.

This allows robust control because the *GA*, combined with feedback, guarantee robustness. However, robust control is not necessarily optimal control. The *GA* attempts to find a global optimum solution for a given solution space. Any random disturbance ($m(t)$ in Fig. 1) of the plant can "kick" the GA into a different solution space.

It is desirable, however, to search for a global optimum in multiple solution spaces in order to find a "universal" optimum.

The application of new knowledge-based control algorithms in advanced intelligent control theory of complex dynamic systems (as in controlling objects) has brought in necessity of the development of new processing methods such as Computational Intelligence (CI). Traditional computing basic tools for CI is GA, FNN, Fuzzy Sets theory, Evolution Programming, Qualitative Probabilistic Reasoning and etc. Application of CI in the advanced control theory of complex system motion has brought to two researching ways: 1) the study of stable non-equilibrium motion; and 2) an unstable non-equilibrium motion of complex dynamic systems.

In the first case (of stable non-equilibrium motion) the development and design of intelligent control algorithms can be described in the structure submitted in Fig. 1.

The peculiarity of the given structure is the consideration of the control object in accordance with the fuzzy system theory as a "black box" or non-linear models of plants. The study and optimization of "input-output" relations is based on soft computing as GA, FNN and fuzzy control (FC) for the description of the changing law of a PID-controller parameters with a minimum entropy production and control error. At the small random initial states, uncontrollable external excitations or small change of parameters or structure of controlled objects (plants), such an approach guarantees a robust and stable control for fixed spaces of possible excitations and solutions.

In case of a global unstable dynamic control objective such an approach based on the presence of a robust and stable control does not guarantee success in principle. For such a kind of unstable dynamic control objectives, the development of new intelligent robust algorithms based on the knowledge about a movement of essentially non-linear unstable non-holonomic dynamic systems is necessary.

## OBJECT AND SUMMARY OF THE INVENTION

It has been found and is the object of the present invention a new method and a respective hardware architecture for controlling a generic process using an

algorithm that is obtained substantially by merging a Genetic Algorithm and a Quantum Search Algorithm.

A hardware structure of a generic control system with a reduced number of sensors is described. Several practical embodiments of such a structure are described. By way of example, the novel structure of the invention is used for implementing a control system with a reduced number of sensors for an internal combustion engine and for a vehicle suspension.

Naturally, the method as well as the hardware architecture of the invention can also be used efficiently for searching data in a database and in similar applications.

Integrated silicon devices containing circuits that implement the different steps of the method of the invention are also described.

The invention is concisely defined in the attached claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

The particular aspects and advantages of the invention will become clearer through the following description of several important embodiments of the and by referring to the attached drawings, wherein:

**Figure 1** is a general structure of an Intelligent Control System based on Soft-Computing;

**Figure 2** is a general structure of an Intelligent Smart Control System, based on Quantum Soft-Computing;

**Figure 3** is a block diagram of Quantum Algorithms;

**Figure 4** is a block diagram of an Encoder;

**Figure 5** is a general structure of the Quantum Block in Fig. 3;

**Figure 6** is an example of Quantum Circuit;

**Figure 7.a** shows an example of tensor product transformation;

**Figure 7.b** shows an example of dot product transformation;

**Figure 7.c** shows the identity transformation;

**Figure 7.d** shows an example of propagation rule;

**Figure 7.e** shows an example of iteration rule;

**Figure 7.f** explains the input/output tensor rule;

**Figure 8** is an example of a circuit realizing Grover's Quantum Gate;

**Figure 9** is a Grover's Quantum Gate;

**Figure 10.a** shows a possible set of input probability amplitudes;

**Figure 10.b** shows a set of probability amplitudes after Step 1 of Fig. 8;

**Figure 10.c** shows a set of probability amplitudes after Step 2 of Fig. 8;

**Figure 10.d** shows a set of probability amplitudes after Step 3 of Fig. 8;

**Figure 11.a** is an example of a vector superposition;

**Figure 11.b** is the set of vectors of Fig. 11.a after ${}^4H$ has been applied;

**Figure 11.c** is the set of vectors of Fig. 11.b after the entanglement operator $U_F$ (with $x$=001) has been applied;

**Figure 11.d** is the set of vectors of Fig. 11.b after having applied the interference operator $D_n \otimes I$;

**Figure 11.e** is the set of vectors of Fig. 11.b after having applied the entanglement operator $U_F$ a second time;

**Figure 11.f** is the set of vectors of Fig. 11.b after having applied the interference operator $D_n \otimes I$ a second time;

**Table 1** resumes the information analysis of Grover's algorithm for a general number of iterations;

**Table 2** resumes the information analysis of Grover's algorithm for the first iteration;

**Table 3** resumes the information analysis of Grover's algorithm for the second iteration;

**Figure 12** shows the analogy between GA and QSA;

**Figure 13** is a block diagram for QGSA;

**Figure 14** is a block diagram of Quantum Genetic Search Algorithm;

**Figure 15** describes the structure of classical Genetic Algorithms and Quantum Searching Algorithms for global optimization;

**Figure 16** describes the general structure of Quantum Algorithms;

**Figure 17** depicts a quantum network for Quantum Searching Algorithm;

**Figure 18** is a diagram of a Quantum Searching Algorithm;

**Figure 19** shows several possible distributions of road profiles;

**Figure 20.a** compares different dynamic behaviour of pitch angle of a vehicle after a simulation of a fuzzy control;

**Figure 20.b** compares the phase portraits of the time diagrams of Fig. 20.a;

**Figure 21.a** compares the entropy accumulation of the corresponding dynamic behaviour of Fig. 20.a;

**Figure 21.b** compares the phase portraits of the time diagrams of Fig. 21.a;

**Figure 22** is a block diagram of the accelerator of the invention;

**Figure 23** is a block diagram of a Quantum Gate implementing the Grover's Algorithm;

**Figure 24** is an example of a Quantum Gate for Deutsch-Jozsa Algorithm;

**Figure 25** is an embodiment of a superposition subsystem for Deutsch-Jozsa Algorithm;

**Figure 26** is an embodiment of an interference subsystem for Deutsch-Jozsa Algorithm;

**Figure 27** is an example of a Quantum Gate for Grover's Algorithm;

**Figure 28** is an embodiment of a superposition subsystem for Grover's Algorithm;

**Figure 29** is an embodiment of an interference subsystem for Grover's Algorithm;

**Figure 30** is an embodiment of a $K^{th}$ entanglement subsystem;

**Figure 31** is an embodiment of a $K^{th}$ interference subsystem;

**Figure 32** is a basic structure of intelligent control system simulator with reduced number of sensors;

**Figure 33** is a more detailed structure of the intelligent control system simulator with reduced number of sensors of Fig. 32;

**Table 4** resumes the meaning of labels in Fig. 33;

**Figure 34** depicts an internal combustion piston engine;

**Figure 35** is a block diagram of a reduced control system for an internal combustion engine not implementing a Quantum Genetic Search Algorithm;
**Figure 36** is a block diagram of a reduced control system for an internal combustion engine implementing a Quantum Genetic Search Algorithm.

## DESCRIPTION OF SEVERAL EMBODIMENT OF THE INVENTION

As stated above, the fundamental aspect of the present invention consists in the intuition that the merging of a Quantum Algorithm with a Genetic Algorithm obtaining what may be defined as a Quantum Genetic Search Algorithm would, as indeed has been found true, enhance formidably the function of teaching a Fuzzy Neural Network in implementing intelligent control systems as well as for searching data in a database.

Such a merging has been possible because of the "similarity" between a Quantum Algorithm and a Genetic Algorithm, and in order to make more easily comprehensible the description of the novel Quantum Genetic Search Algorithm of the invention, it is deemed useful to preliminarily present a brief review of the Genetic Algorithm theory and of the Quantum Algorithm theory.

### GENETIC ALGORITHMS

Genetic Algorithms (GA's) are global search algorithms based on the mechanics of natural genetics and natural selection. In the genetic search, a finite length binary string represents each design variable and the set of all possible solutions is so encoded into a population of binary strings. Genetic transformations, analogous to biological reproduction and evolution, are subsequently used to vary and improve the encoded solutions. Usually, three main operators i.e. crossover, mutation and selection are used in the genetic search.

The selection process is one that biases the search toward producing more fit members in the population and eliminating the less fit ones. Hence, a fitness value is first assigned to each string in the population. One simple approach to select

members from an initial population is to assign each member a probability of being selected, on the basis of its fitness value. A new population pool of the same size as the original is then created with a higher average fitness value.

The process of selection simply results in more copies of the dominant design to be present in the population. The crossover process allows for an exchange of design characteristics among members of the population pool with the intent of improving the fitness of the next generation. Crossover is executed by selecting strings of two mating parents, randomly choosing two sites on the strings, and swapping strings of 0's and 1's between these chosen sites.

Mutation safeguards the genetic search process from a premature loss of valuable genetic material during selection and crossover. The process of mutation consists into choosing a few members from the population pool on the basis of their probability of mutation and switch 0 to 1 or vice versa at a randomly selected mutation rate on the selected string.

In the foregoing discussion, the mechanics of the genetic search are simple. However, there are some key differences from traditional methods that contribute to the strength of the approach. GA's work on function evaluations alone and do not require function derivatives. While derivatives contribute to a faster convergence towards the optimum, they may also direct the search towards a local optimum. Furthermore, since the search proceeds from several points in the design space to another set of design points, the method has a higher probability of locating a global minimum as opposed to those schemes that proceed from one point to another. In addition, genetic algorithms work on a coding of design variables rather than variables themselves. This allows for an extension of these algorithms to design space consisting of a mix of continuous, discrete and integer variables.

In the context of GAs operating on binary strings, a schema (or similarity template) is a string of symbols taken from the alphabet $\{0,1,\#\}$ . The character # is interpreted as "don't care" symbol, so that a schema can represent several bit

strings. For example, the schema #10#1 represents four strings: 10011,01011,11001, and 11011. The number of non-# symbols is called the *order* $O(H)$ of a schema $H$. The distance between the furthest two non-# symbols is called the *defining length* $L(H)$ of the schema.

Holland obtained a result (the schema theorem) that predicts how the number of strings in a population matching (or belonging to) a schema is expected to vary from one generation to the next (Holland, 1992). The theorem is as follows:

$$E[m(H,t+1)] \geq m(H,t) \cdot \underbrace{\frac{f(H,t)}{\bar{f}(t)}}_{Selection} \cdot \underbrace{(1-p_m)^{O(H)}}_{Mutation} \cdot \underbrace{[1-p_c \overbrace{\frac{L(H)}{N-1}(1-\frac{m(H,t)f(H,t)}{M\bar{f}(t)})}^{P_d(H,t)}]}_{Crossover} \quad (1)$$

where $m(H,t)$ is the number of strings matching the schema $H$ at generation $t$, $f(H,t)$ is the mean fitness of the strings matching $H$, $\bar{f}(t)$ is the mean fitness of the strings in the population, $p_m$ is the probability of mutation per bit, $p_c$ is the probability of crossover, $N$ is the number of bits in the strings, $M$ is the number of strings in the population, and $E[m(H,t+1)]$ is the expected number of strings matching the schema $H$ at generation $t+1$. This is slightly different version of Holland's original theorem. Equation (1) applies when crossover is performed taking both parents from the mating pool (Goldberg, 1989). The three horizontal curly brackets beneath the equation indicate which operators are responsible for each term. The bracket above the equation represents the probability of disruption of the schema $H$ at generation $t$ due to crossover $P_d(H,t)$. Such a probability depends on the frequency of the schema in the mating pool but also on the intrinsic *fragility* of the schema $L(H)/(N-1)$.

**QUANTUM ALGORITHMS**

The problems solved by the quantum algorithms may be stated as follows:

| | |
|---|---|
| **Input** | A function $f:\{0,1\}^n \to \{0,1\}^m$ |
| **Problem** | Find a certain property of $f$ |

The structure of a quantum algorithm is outlined, by a high level representation, in the schematic diagram of Fig. 3.

The input of a quantum algorithm is always a function $f$ from binary strings into binary strings. This function is represented as a map table, defining for every string its image. Function $f$ is firstly encoded into a unitary matrix operator $U_F$ depending on $f$ properties. In some sense, this operator calculates $f$ when its input and output strings are encoded into canonical basis vectors of a Complex Hilbert Space: $U_F$ maps the vector code of every string into the vector code of its image by $f$.

BOX 1: UNITARY MATRIX $U_F$

A squared matrix $U_F$ on the complex field is *unitary* if its inverse matrix coincides with its conjugate transpose:

$$U_F^{-1}=U_F^{\dagger}$$

A unitary matrix is always reversible and preserves the norm of vectors.

When the matrix operator $U_F$ has been generated, it is embedded into a quantum gate $G$, a unitary matrix whose structure depends on the form of matrix $U_F$ and on the problem we want to solve. The quantum gate is the heart of a quantum algorithm. In every quantum algorithm, the quantum gate acts on an initial canonical basis vector (we can always choose the same vector) in order to generate a complex linear combination (let's call it superposition) of basis vectors as output. This superposition contains all the information to answer the initial problem.

After this superposition has been created, measurement takes place in order to extract this information. In quantum mechanics, measurement is a non-deterministic operation that produces as output only one of the basis vectors in the entering superposition. The probability of every basis vector of being the output of measurement depends on its complex coefficient (probability amplitude) in the entering complex linear combination.

The segmental action of the quantum gate and of measurement constitutes the quantum block. The quantum block is repeated $k$ times in order to produce a collection of $k$ basis vectors. Being measurement a non-deterministic operation, these basic vectors won't be necessarily identical and each one of them will encode a peace of the information needed to solve the problem.

The last part of the algorithm consists into the interpretation of the collected basis vectors in order to get the right answer for the initial problem with a certain probability.

## Encoder

The behaviour of the encoder block is described in the detailed schematic diagram of Fig. 4.

Function $f$ is encoded into matrix $U_F$ in three steps.

*Step 1*

The map table of function $f:\{0,1\}^n \rightarrow \{0,1\}^m$ is transformed into the map table of the injective function $F:\{0,1\}^{n+m} \rightarrow \{0,1\}^{n+m}$ such that:

$$F(x_0, .., x_{n-1}, y_0, .., y_{m-1}) = (x_0, .., x_{n-1}, f(x_0, .., x_{n-1}) \oplus (y_0, .., y_{m-1})) \qquad (2)$$

The need to deal with an injective function comes from the requirement that $U_F$ is unitary. A unitary operator is reversible, so it can't map 2 different inputs in the same output. Since $U_F$ will be the matrix representation of $F$, $F$ is supposed to be injective. If we directly employed the matrix representation of function $f$, we

could obtain a non-unitary matrix, since $f$ could be non-injective. So, injectivity is fulfilled by increasing the number of bits and considering function $F$ instead of function $f$. Anyway, function $f$ can always be calculated from $F$ by putting $(y_0, .., y_{m-1})=(0,..,0)$ in the input string and reading the last $m$ values of the output string.

> BOX 2: XOR OPERATOR $\oplus$
>
> The XOR operator between two binary strings $p$ and $q$ of length $m$ is a string $s$ of length $m$ such that the $i$-th digit of $s$ is calculated as the exclusive OR between the $i$-th digits of $p$ and $q$:
>
> $$p=(p_0, .., p_{n-1})$$
>
> $$q=(q_0, .., q_{n-1})$$
>
> $$s = p \oplus q = ((p_0+q_0) \bmod 2, .., (p_{n-1}+q_{n-1}) \bmod 2))$$

<u>*Step 2*</u>

Function $F$ map table is transformed into $U_F$ map table, following the following constraint:

$$\forall s \in \{0,1\}^{n+m} : U_F[\tau(s)] = \tau[F(s)] \qquad (3)$$

The code map $\tau : \{0,1\}^{n+m} \rightarrow \mathbf{C}^{2^{n+m}}$ ($\mathbf{C}^{2^{n+m}}$ is the target Complex Hilbert Space) is such that:

$$\tau(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \quad \tau(1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$
$$\tau(x_0,..,x_{n+m-1}) = \tau(x_0) \otimes .. \otimes \tau(x_{n+m-1}) = |x_0..x_{n+m-1}\rangle$$

Code $\tau$ maps bit values into complex vectors of dimension 2 belonging to the canonical basis of $\mathbf{C}^2$. Besides, using tensor product, $\tau$ maps the general state of a

binary string of dimension $n$ into a vector of dimension $2^n$, reducing this state to the joint state of the $n$ bits composing the register. Every bit state is transformed into the corresponding 2-dimesional basis vector and then the string state is mapped into the corresponding $2^n$-dimesional basis vector by composing all bit-vectors through tensor product. In this sense tensor product is the vector counterpart of state conjunction.

BOX 3: VECTOR TENSOR PRODUCT ⊗

The tensor product between two vectors of dimensions $h$ and $k$ is a tensor product of dimension $h \cdot k$, such that:

$$|x\rangle \otimes |y\rangle = \begin{pmatrix} x_1 \\ \dots \\ x_h \end{pmatrix} \otimes \begin{pmatrix} y_1 \\ \dots \\ y_k \end{pmatrix} = \begin{pmatrix} x_1 y_1 \\ \dots \\ x_1 y_k \\ \dots \\ x_h y_1 \\ \dots \\ x_h y_k \end{pmatrix} \Rightarrow$$

**Physical interpretation:**

*If a component of a complex vector is interpreted as the probability amplitude of a system of being in a given state (indexed by the component number), the tensor product between two vectors describes the joint probability amplitude of two systems of being in a joint state.*

**Examples: Vector Tensor Products**

$$(0,0) \xrightarrow{\tau} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |00\rangle$$

$$(0,1) \xrightarrow{\tau} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle$$

| | |
|---|---|
| $(1,0) \xrightarrow{\tau} \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} = \lvert 10\rangle$ | $(1,1) \xrightarrow{\tau} \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \lvert 11\rangle$ |

Basis vectors are denoted using the *ket* notation $|i>$. This notation is taken from Dirac description of quantum mechanics.

<u>*Step 3*</u>

$U_F$ map table is transformed into $U_F$ using the following transformation rule:

$$[U_F]_{ij} = 1 \Leftrightarrow U_F|j\rangle = |i\rangle \qquad (4)$$

This rule can easily be understood considering vectors $|i>$ and $|j>$ as column vectors. Belonging these vectors to the canonical basis, $U_F$ defines a <u>permutation map</u> of the <u>identity matrix</u> rows. In general, row $|j>$ is mapped into row $|i>$.

This rule will be illustrated in detail on the first example of quantum algorithm: Grover's algorithm.

<u>**Quantum block**</u>

The heart of the quantum block is the <u>quantum gate</u>, which depends on the properties of matrix $U_F$. The scheme in Fig. 5 gives a more detailed description of the quantum block.

The matrix operator $U_F$ in Fig. 5 is the output of the encoder block represented in Fig. 4. Here, it becomes the input for the quantum block.

This matrix operator is firstly embedded into a more complex gate: the quantum gate $G$. Unitary matrix $G$ is applied $k$ times to an initial canonical basis vector $|i>$ of dimension $2^{n+m}$. Every time, the resulting complex superposition $G|0..01..1>$ of basis vectors is measured, producing one basis vector $|x_i>$ as result. All the

measured basis vectors $\{|x_1\rangle,..,|x_k\rangle\}$ are collected together. This collection is the output of the quantum block.

The "intelligence" of such algorithms is in the ability to build a quantum gate that is able to extract the information necessary to find the required property of $f$ and to store it into the output vector collection.

The structure of the quantum gate for every quantum algorithm will be discussed in detail, observing that a general description is possible.

In order to represent quantum gates we are going to employ some special diagrams called quantum circuits.

An example of quantum circuit is reported in Fig. 6. Every rectangle is associated to a matrix $2^n \times 2^n$, where $n$ is the number of lines entering and leaving the rectangle. For example, the rectangle marked $U_F$ is associated to matrix $U_F$.

Quantum circuits let us give a high-level description of the gate and, using some transformation rules, which are listed in Fig. 7, it is possible to compile them into the corresponding gate-matrix.

It will be clearer how to use these rules when we afford the first examples of quantum algorithm.

BOX 4: MATRIX TENSOR PRODUCT ⊗

The tensor product between two matrices $X_{n\times m}$ and $Y_{h\times k}$ is a (block) matrix $(n \cdot h)\times(m \cdot k)$ such that:

$$X \otimes Y = \begin{bmatrix} x_{11}Y & .. & x_{1m}Y \\ .. & .. & .. \\ x_{n1}Y & .. & x_{nm}Y \end{bmatrix} \quad with \quad X = \begin{bmatrix} x_{11} & .. & x_{1m} \\ .. & .. & .. \\ x_{n1} & .. & x_{nm} \end{bmatrix}$$

**Example: Matrix Tensor Product**

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\otimes\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}=\begin{bmatrix}1\cdot\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix} & 2\cdot\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}\\3\cdot\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix} & 4\cdot\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}\end{bmatrix}=\begin{bmatrix}5 & 6 & 10 & 12\\7 & 8 & 14 & 16\\15 & 18 & 20 & 24\\21 & 24 & 28 & 32\end{bmatrix}$$

**Decoder**

The decoder block has the function to interpret the basis vectors collected after the iterated execution of the quantum block. Decoding these vectors means to retranslate them into binary strings and interpreting them directly if they already contain the answer to the starting problem or use them, for instance as coefficients vectors for some equation system, in order to get the searched solution. This part will not being investigated in detail because it is a non-interesting easy classical part.

Because of the particular importance of the Grover's Algorithm in the realization of controllers, a brief description of the Grover's algorithm is given.

**GROVER'S PROBLEM**

Grover's problem is so stated:

| | |
|---|---|
| **Input** | A function $f:\{0,1\}^n \rightarrow \{0,1\}$ such that $\exists x\in\{0,1\}^n$: $(f(x)=1 \wedge \forall y\in\{0,1\}^n: x\neq y \Rightarrow f(y)=0)$ |
| **Problem** | Find $x$ |

In Deutsch-Jozsa's algorithm we distinguished two classes of input functions and we were supposed to decide what class the input function belonged to. In this case the problem is in some sense identical in its form, even if it is harder because now we are dealing with $2^n$ classes of input functions (each function of the kind described constitutes a class).

## Encoder

In order to make the discussion more comprehensible, we prefer firstly to consider a special function with $n$=2. Then we discuss the general case with $n$=2 and finally we analyze the general case with $n$>0.

*A1. Introductory example*

Let's consider the case:

$$n = 2 \qquad f(01) = 1$$

In this case $f$ map table is so defined:

| $x$ | $f(x)$ |
|---|---|
| 00 | 0 |
| 01 | 1 |
| 10 | 0 |
| 11 | 0 |

*Step 1*

Function $f$ is encoded into injective function $F$, built according to the usual statement:

$$F:\{0,1\}^{n+1} \to \{0,1\}^{n+1} : F(x_0, x_1, y_0) = (x_0, x_1, f(x_0, x_1) \oplus y_0)$$

Then $F$ map table is:

| $(x_0, x_1, y_0)$ | $F(x_0, x_1, y_0)$ |
|---|---|
| 000 | 000 |
| 010 | 011 |
| 100 | 100 |
| 110 | 110 |
| 001 | 001 |
| 011 | 010 |
| 101 | 101 |
| 111 | 111 |

*Step 2*

Let's now encode $F$ into the map table of $U_F$ using the usual rule:

$$\forall s \in \{0,1\}^{n+1}: U_F[\tau(s)] = \tau[F(s)]$$

where $\tau$ is the code map defined in Equation (3). This means:

| $\lvert x_0\, x_1\, y_0>$ | $U_F \lvert x_0\, x_1\, y_0>$ |
|---|---|
| \|000> | \|000> |
| \|010> | \|011> |
| \|100> | \|100> |
| \|110> | \|110> |
| \|001> | \|001> |
| \|011> | \|011> |
| \|101> | \|101> |
| \|111> | \|111> |

*Step 3*

From the map table of $U_F$ we are supposed to calculate the corresponding matrix operator.

This matrix is obtained from Equation (4) using the rule:

$$[U_F]_{ij} = 1 \Leftrightarrow U_F |j\rangle = |i\rangle$$

$U_F$ is so calculated:

| $U_F$ | \|00> | \|01> | \|10> | \|11> |
|---|---|---|---|---|
| \|00> | **I** | 0 | 0 | 0 |
| \|01> | 0 | **C** | 0 | 0 |
| \|10> | 0 | 0 | **I** | 0 |
| \|11> | 0 | 0 | 0 | **I** |

The effect of this matrix is to leave unchanged the first and the second input basis vectors of the input tensor product, flipping the third one when the first vector is |0> and the second is |1>. This agrees with the constraints on $U_F$ stated above.

*B. General case with n=2*

Let's now take into consideration the more general case:

$$n=2 \qquad f(\underline{x})=1$$

The corresponding matrix operator is:

| $U_F$ | \|00> | \|01> | \|10> | \|11> |
|---|---|---|---|---|
| \|00> | $M_{00}$ | 0 | 0 | 0 |
| \|01> | 0 | $M_{01}$ | 0 | 0 |
| \|10> | 0 | 0 | $M_{10}$ | 0 |
| \|11> | 0 | 0 | 0 | $M_{11}$ |

with $M_{\underline{x}}=C\wedge\forall i\neq\underline{x}:M_i=I$.

*C. General case*

It is fairly natural to generalize operator $U_F$ from the case $n$=2 to the case $n$>1. In fact, we always find operator $C$ on the main diagonal of the block matrix, in correspondence of the celled labeled by vector $|\underline{x}>$, where $\underline{x}$ is the binary string having image one by $f$. Therefore:

| $U_F$ | \|00> | \|01> | ... | \|11> |
|---|---|---|---|---|
| \|00> | $M_{00}$ | 0 | ... | 0 |
| \|01> | 0 | $M_{01}$ | ... | 0 |
| ... | ... | ... | ... | ... |
| \|11> | 0 | 0 | ... | $M_{11}$ |

with $M_{\underline{x}}=C\wedge\forall i\neq\underline{x}:M_i=I$.

Matrix $U_F$, the output of the encoder, is embedded into the quantum gate. We describe this gate using a quantum circuit such that depicted in Fig. 8.

Operator $D_n$ is called <u>diffusion matrix</u> of order $n$ and it is responsible of interference in this algorithm. It plays the same role as $QFT_n$ in Shor's algorithm and of ${}^nH$ in Deutsch-Jozsa's and Simon's algorithms. This matrix is defined in this way:

| $D_n$ | \|0..0> | \|0..1> | ... | \|i> | ... | \|1..0> | \|1..1> |
|---|---|---|---|---|---|---|---|
| \|0..0> | **$-1+1/2^{n-1}$** | $1/2^{n-1}$ | ... | $1/2^{n-1}$ | ... | $1/2^{n-1}$ | $1/2^{n-1}$ |
| \|0..1> | $1/2^{n-1}$ | **$-1+1/2^{n-1}$** | ... | $1/2^{n-1}$ | ... | $1/2^{n-1}$ | $1/2^{n-1}$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| \|i> | $1/2^{n-1}$ | $1/2^{n-1}$ | ... | **$-1+1/2^{n-1}$** | ... | $1/2^{n-1}$ | $1/2^{n-1}$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| \|1..0> | $1/2^{n-1}$ | $1/2^{n-1}$ | ... | $1/2^{n-1}$ | ... | **$-1+1/2^{n-1}$** | $1/2^{n-1}$ |
| \|1..1> | $1/2^{n-1}$ | $1/2^{n-1}$ | ... | $1/2^{n-1}$ | ... | $1/2^{n-1}$ | **$-1+1/2^{n-1}$** |

Using the identity transformation (Fig. 7.c), the previous circuit can be compiled into the circuit of Fig. 9.

*A2. Introductory example: Dynamic Analysis*

In the introductory example we dealt above, $U_F$ had the following form:

| $U_F$ | \|00> | \|01> | \|10> | \|11> |
|---|---|---|---|---|
| \|00> | **I** | 0 | 0 | 0 |
| \|01> | 0 | **C** | 0 | 0 |
| \|10> | 0 | 0 | **I** | 0 |
| \|11> | 0 | 0 | 0 | **I** |

Let's calculate the quantum gate $G=[(D_2 \otimes I) \cdot U_F]^h \cdot (^{2+1}H)$ in this case:

| $^3H$ | \|00> | \|01> | \|10> | \|11> |
|---|---|---|---|---|
| \|00> | $H/2$ | $H/2$ | $H/2$ | $H/2$ |
| \|01> | $H/2$ | $-H/2$ | $H/2$ | $-H/2$ |
| \|10> | $H/2$ | $H/2$ | $-H/2$ | $-H/2$ |
| \|11> | $H/2$ | $-H/2$ | $-H/2$ | $H/2$ |

| $D_2 \otimes I$ | \|00> | \|01> | \|10> | \|11> |
|---|---|---|---|---|
| \|00> | $-I/2$ | $I/2$ | $I/2$ | $I/2$ |
| \|01> | $I/2$ | $-I/2$ | $I/2$ | $I/2$ |
| \|10> | $I/2$ | $I/2$ | $-I/2$ | $I/2$ |
| \|11> | $I/2$ | $I/2$ | $I/2$ | $-I/2$ |

| $U_F \cdot {}^3H$ | \|00> | \|01> | \|10> | \|11> |
|---|---|---|---|---|
| \|00> | $H/2$ | $H/2$ | $H/2$ | $H/2$ |
| \|01> | $CH/2$ | $-CH/2$ | $CH/2$ | $-CH/2$ |
| \|10> | $H/2$ | $H/2$ | $-H/2$ | $-H/2$ |
| \|11> | $H/2$ | $-H/2$ | $-H/2$ | $H/2$ |

Choosing $h$=1, we obtain:

| G | \|00> | \|01> | \|10> | \|11> |
|---|---|---|---|---|
| \|00> | $(C+I)H/4$ | $(-C-I)H/4$ | $(C-3I)H/4$ | $(-C-I)H/4$ |
| \|01> | $(-C+3I)H/4$ | $(C+I)H/4$ | $(-C-I)H/4$ | $(C+I)H/4$ |
| \|10> | $(C+I)H/4$ | $(-C-I)H/4$ | $(C+I)H/4$ | $(-C+3I)H/4$ |
| \|11> | $(C+I)H/4$ | $(-C+3I)H/4$ | $(C+I)H/4$ | $(-C-I)H/4$ |

Now, consider the application of G to vector |001>:

$$G|001\rangle = \frac{1}{4}|00\rangle \otimes (C+I)H|1\rangle + \frac{1}{4}|01\rangle \otimes (-C+3I)H|1\rangle +$$
$$\frac{1}{4}|10\rangle \otimes (C+I)H|1\rangle + \frac{1}{4}|11\rangle \otimes (C+I)H|1\rangle$$

Let's calculate the operator $(-C+3I)H/4$. Then

| **-C+3I** | \|0> | \|1> |
|---|---|---|
| \|0> | 3 | -1 |
| \|1> | -1 | 3 |

| **(-C+3I)H/4** | \|0> | \|1> |
|---|---|---|
| \|0> | $1/2^{3/2}$ | $1/2^{1/2}$ |
| \|1> | $1/2^{3/2}$ | $-1/2^{1/2}$ |

Therefore:

$$\frac{1}{4}(-C+3I)H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Let's calculate the operator $(C+I)H/4$. Then:

| **C+I** | \|0> | \|1> |
|---|---|---|
| \|0> | 1 | 1 |
| \|1> | 1 | 1 |

| **(C+I)H/4** | \|0> | \|1> |
|---|---|---|
| \|0> | $1/2^{3/2}$ | 0 |
| \|1> | $1/2^{3/2}$ | 0 |

Therefore:

$$\frac{1}{4}(C+I)H|1\rangle = 0$$

This means that |001> is mapped into vector |01>(|0>-|1>)/$2^{1/2}$. Taking the binary values of the first two vectors of dimension 2, we find $\underline{x}$.

It might be useful to picture the evolution of the probability amplitude of every basis vector while operator ${}^3H$. $U_F$ and $D_2 \otimes I$ are applied in sequence. This is done in Fig. 10.

Operator ${}^3H$ puts the initial canonical basis vector |001> into a superposition of all basis vectors with the same (real) coefficients in modulus, but with positive sign if the last vector is |0>, negative otherwise. Operator $U_F$ creates <u>correlation</u>: it flips the third vector if the first two vector are |0> and |1>. Finally, $D_2 \otimes I$ produces <u>interference</u>: for every basis vector $|\underline{x}_0\underline{x}_1\underline{y}_0\rangle$ it calculates its output probability amplitude $\alpha'_{x_0x_1y_0}$ by inverting its initial probability amplitude $\alpha_{x_0x_1y_0}$ and summing the double of the mean $\underline{\alpha}_{y_0}$ of the probability amplitude of all vectors in the form $|x_0x_1\underline{y}_0\rangle$. In our example $\underline{\alpha}_0=1/(4\cdot 2^{1/2})$, $\underline{\alpha}_1= -1/(4\cdot 2^{1/2})$. Take, for instance, basis vector |000>. Then $\alpha'_{000}=-\alpha_{000}+2\underline{\alpha}_0=-1/(2\cdot 2^{1/2})+2/(4\cdot 2^{1/2})=0$.

*<u>D. General case with n=2</u>*

In general, if $n$=2, $U_F$ has the following form:

| $U_F$ | \|00> | \|01> | \|10> | \|11> |
|---|---|---|---|---|
| \|00> | $M_{00}$ | 0 | 0 | 0 |
| \|01> | 0 | $M_{01}$ | 0 | 0 |
| \|10> | 0 | 0 | $M_{10}$ | 0 |
| \|11> | 0 | 0 | 0 | $M_{11}$ |

where $M_{\underline{x}}= C \wedge \forall i \neq \underline{x}: M_i = I$ ($\underline{x}, i \in :\{0,1\}^n$ ).

Let's calculate the quantum gate $G=(D_2 \otimes I)\cdot U_F \cdot ({}^{2+1}H)$ in this general case:

| $U_F \cdot {}^3H$ | $\|00>$ | $\|01>$ | $\|10>$ | $\|11>$ |
|---|---|---|---|---|
| $\|00>$ | $M_{00}H/2$ | $M_{00}H/2$ | $M_{00}H/2$ | $M_{00}H/2$ |
| $\|01>$ | $M_{01}H/2$ | $-M_{01}H/2$ | $M_{01}H/2$ | $-M_{01}H/2$ |
| $\|10>$ | $M_{10}H/2$ | $M_{10}H/2$ | $-M_{10}H/2$ | $-M_{10}H/2$ |
| $\|11>$ | $M_{11}H/2$ | $-M_{11}H/2$ | $-M_{11}H/2$ | $M_{11}H/2$ |

⇓

| G | $\|00>$ | $\|01>$ |
|---|---|---|
| $\|00>$ | $(-M_{00}+M_{01}+M_{10}+M_{11})H/4$ | $(-M_{00}-M_{01}+M_{10}-M_{11})H/4$ |
| $\|01>$ | $(M_{00}-M_{01}+M_{10}+M_{11})H/4$ | $(M_{00}+M_{01}+M_{10}-M_{11})H/4$ |
| $\|10>$ | $(M_{00}+M_{01}-M_{10}+M_{11})H/4$ | $(M_{00}-M_{01}-M_{10}-M_{11})H/4$ |
| $\|11>$ | $(M_{00}+M_{01}+M_{10}-M_{11})H/4$ | $(M_{00}-M_{01}+M_{10}+M_{11})H/4$ |

| G | $\|10>$ | $\|11>$ |
|---|---|---|
| $\|00>$ | $(-M_{00}+M_{01}-M_{10}-M_{11})H/4$ | $(-M_{00}-M_{01}-M_{10}+M_{11})H/4$ |
| $\|01>$ | $(M_{00}-M_{01}-M_{10}-M_{11})H/4$ | $(M_{00}+M_{01}-M_{10}+M_{11})H/4$ |
| $\|10>$ | $(M_{00}+M_{01}+M_{10}-M_{11})H/4$ | $(M_{00}-M_{01}+M_{10}+M_{11})H/4$ |
| $\|11>$ | $(M_{00}+M_{01}-M_{10}+M_{11})H/4$ | $(M_{00}-M_{01}-M_{10}-M_{11})H/4$ |

Now, consider the application of $G$ to vector $|001>$:

$$G|001\rangle = \frac{1}{4}|00\rangle \otimes (-M_{00}+M_{01}+M_{10}+M_{11})H|1\rangle + \frac{1}{4}|01\rangle \otimes (M_{00}-M_{01}+M_{10}+M_{11})H|1\rangle + \frac{1}{4}|10\rangle \otimes (M_{00}+M_{01}-M_{10}+M_{11})H|1\rangle + \frac{1}{4}|11\rangle \otimes (M_{00}+M_{01}+M_{10}-M_{11})H|1\rangle$$

Consider the following cases:

**$\underline{x}$=00:**

$$G|001\rangle = \frac{1}{4}|00\rangle \otimes (-C+3I)H|1\rangle + \frac{1}{4}|01\rangle \otimes (C+I)H|1\rangle + \frac{1}{4}|10\rangle \otimes (C+I)H|1\rangle + \frac{1}{4}|11\rangle \otimes (C+I)H|1\rangle = |00\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

**$\underline{x}$=01:**

$$G|001\rangle = \frac{1}{4}|00\rangle \otimes (C+I)H|1\rangle + \frac{1}{4}|01\rangle \otimes (-C+3I)H|1\rangle +$$
$$\frac{1}{4}|10\rangle \otimes (C+I)H|1\rangle + \frac{1}{4}|11\rangle \otimes (C+I)H|1\rangle = |01\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

**$\underline{x}$=10:**

$$G|001\rangle = \frac{1}{4}|00\rangle \otimes (C+I)H|1\rangle + \frac{1}{4}|01\rangle \otimes (C+I)H|1\rangle +$$
$$\frac{1}{4}|10\rangle \otimes (-C+3I)H|1\rangle + \frac{1}{4}|11\rangle \otimes (C+I)H|1\rangle = |10\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

**$\underline{x}$=11:**

$$G|001\rangle = \frac{1}{4}|00\rangle \otimes (C+I)H|1\rangle + \frac{1}{4}|01\rangle \otimes (C+I)H|1\rangle +$$
$$\frac{1}{4}|10\rangle \otimes (C+I)H|1\rangle + \frac{1}{4}|11\rangle \otimes (-C+3I)H|1\rangle = |11\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

This means that if we measure the output vector and encode back the first two basis vectors of dimension 2 in the resulting tensor product, we get the following results:

| $\underline{x}$ | **Result** | **Probability** |
|---|---|---|
| 00 | 00 | 1 |
| 01 | 01 | 1 |
| 10 | 10 | 1 |
| 11 | 11 | 1 |

_E. General case (n>0)_

In the general case $n$>0, $U_F$ has the following form:

| $U_F$ | \|0..0> | \|0..1> | ... | \|1..1> |
|---|---|---|---|---|
| \|0..0> | $M_{0..0}$ | 0 | 0 | 0 |
| \|0..1> | 0 | $M_{0..1}$ | 0 | 0 |
| ... | ... | ... | ... | ... |
| \|1..1> | 0 | 0 | 0 | $M_{1..1}$ |

where $M_{\underline{x}}= C \wedge \forall i \neq \underline{x}: M_i = I$ ($\underline{x}, i \in : \{0,1\}^n$ ).

Let's calculate the quantum gate $G=(D_n \otimes I)^h \cdot U_F \cdot ({}^{n+1}H)$:

| ${}^{n+1}H$ | $\vert 0..0\rangle$ | ... | $\vert j\rangle$ | ... | $\vert 1..1\rangle$ |
|---|---|---|---|---|---|
| $\vert 0..0\rangle$ | $H/2^{n/2}$ | ... | $H/2^{n/2}$ | ... | $H/2^{n/2}$ |
| ... | ... | ... | ... | ... | ... |
| $\vert i\rangle$ | $H/2^{n/2}$ | ... | $(-1)^{i \cdot j} H/2^{n/2}$ | ... | $(-1)^{i \cdot (1..1)} H/2^{n/2}$ |
| ... | ... | ... | ... | ... | ... |
| $\vert 1..1\rangle$ | $H/2^{n/2}$ | ... | $(-1)^{(1..1) \cdot j} H/2^{n/2}$ | ... | $(-1)^{(1..1) \cdot (1..1)} H/2^{n/2}$ |

| $D_n \otimes I$ | $\vert 0..0\rangle$ | $\vert 0..1\rangle$ | ... | $\vert i\rangle$ | ... | $\vert 1..0\rangle$ | $\vert 1..1\rangle$ |
|---|---|---|---|---|---|---|---|
| $\vert 0..0\rangle$ | $\boldsymbol{-I+I/2^{n-1}}$ | $I/2^{n-1}$ | ... | $I/2^{n-1}$ | ... | $I/2^{n-1}$ | $I/2^{n-1}$ |
| $\vert 0..1\rangle$ | $I/2^{n-1}$ | $\boldsymbol{-I+I/2^{n-1}}$ | ... | $I/2^{n-1}$ | ... | $I/2^{n-1}$ | $I/2^{n-1}$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $\vert i\rangle$ | $I/2^{n-1}$ | $I/2^{n-1}$ | ... | $\boldsymbol{-I+I/2^{n-1}}$ | ... | $I/2^{n-1}$ | $I/2^{n-1}$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $\vert 1..0\rangle$ | $I/2^{n-1}$ | $I/2^{n-1}$ | ... | $I/2^{n-1}$ | ... | $\boldsymbol{-I+I/2^{n-1}}$ | $I/2^{n-1}$ |
| $\vert 1..1\rangle$ | $I/2^{n-1}$ | $I/2^{n-1}$ | ... | $I/2^{n-1}$ | ... | $I/2^{n-1}$ | $\boldsymbol{-I+I/2^{n-1}}$ |

| $U_F \cdot {}^{n+1}H$ | $\vert 0..0\rangle$ | ... | $\vert j\rangle$ | ... | $\vert 1..1\rangle$ |
|---|---|---|---|---|---|
| $\vert 0..0\rangle$ | $M_{0..0}H/2^{n/2}$ | ... | $M_{0..0}H/2^{n/2}$ | ... | $M_{0..0}H/2^{n/2}$ |
| ... | ... | ... | ... | ... | ... |
| $\vert i\rangle$ | $M_i H/2^{n/2}$ | ... | $(-1)^{i \cdot j} M_i H/2^{n/2}$ | ... | $(-1)^{i \cdot (1..1)} M_i H/2^{n/2}$ |
| ... | ... | ... | ... | ... | ... |
| $\vert 1..1\rangle$ | $M_{1..1}H/2^{n/2}$ | ... | $(-1)^{(1..1) \cdot j} M_{1..1}H/2^{n/2}$ | ... | $(-1)^{(1..1) \cdot (1..1)} M_{1..1}H/2^{n/2}$ |

Now, suppose $h$=1. Then:

| $G_{h=1}$ | $\vert 0..0\rangle$ | ... |
|---|---|---|
| $\vert 0..0\rangle$ | $(-M_{0..0}+\Sigma_{j \in \{0,1\}^n} M_j/2^{n-1})H/2^{n/2}$ | ... |
| ... | ... | ... |
| $\vert i\rangle$ | $(-M_i+\Sigma_{j \in \{0,1\}^n} M_j/2^{n-1})H/2^{n/2}$ | ... |
| ... | ... | ... |
| $\vert 1..1\rangle$ | $(-M_{1..1}+\Sigma_{j \in \{0,1\}^n} M_j/2^{n-1})H/2^{n/2}$ | ... |

Being $M_{\underline{x}} = C$ and $\forall i \neq \underline{x}: M_i = I$, this column may be written as:

| $G_{h=1}$ | $\vert 0..0\rangle$ | ... |
|---|---|---|
| $\vert 0..0\rangle$ | $(-I+\Sigma_{j\in\{0,1\}^n-\{\underline{x}\}}\ I/2^{n-1}+C/2^{n-1})H/2^{n/2}$ | ... |
| ... | ... | ... |
| $\vert \underline{x}\rangle$ | $(-C+\Sigma_{j\in\{0,1\}^n-\{\underline{x}\}}\ I/2^{n-1}+C/2^{n-1})H/2^{n/2}$ | ... |
| ... | ... | ... |
| $\vert 1..1\rangle$ | $(-I+\Sigma_{j\in\{0,1\}^n-\{\underline{x}\}}\ I/2^{n-1}+C/2^{n-1})H/2^{n/2}$ | ... |

and so:

| $G_{h=1}$ | $\vert 0..0\rangle$ | ... |
|---|---|---|
| $\vert 0..0\rangle$ | $\{[-1+(2^n-1)/2^{n-1}]I+C/2^{n-1}\}H/2^{n/2}$ | ... |
| ... | ... | ... |
| $\vert \underline{x}\rangle$ | $\{(2^n-1)/2^{n-1}I+[-1+1/2^{n-1}]C\}H/2^{n/2}$ | ... |
| ... | ... | ... |
| $\vert 1..1\rangle$ | $\{[-1+(2^n-1)/2^{n-1}]I+C/2^{n-1}\}H/2^{n/2}$ | ... |

Now, consider to apply matrix operator $\{[-1+(2^n-1)/2^{n-1}]I+C/2^{n-1}\}H/2^{n/2}$ and matrix operator $\{(2^n-1)/2^{n-1}I+[-1+1/2^{n-1}]C\}H/2^{n/2}$ to vector $|1\rangle$:

$$\frac{1}{2^{n/2}}\left\{\left[-1+\frac{2^n-1}{2^{n-1}}\right]I+\frac{1}{2^{n-1}}C\right\}H|1\rangle=\left(-1+\frac{2^n-2}{2^{n-1}}\right)\frac{|0\rangle-|1\rangle}{2^{\frac{(n+1)}{2}}}$$

$$\frac{1}{2^{n/2}}\left\{\frac{2^n-1}{2^{n-1}}I+\left[-1+\frac{1}{2^{n-1}}\right]C\right\}H|1\rangle=\left(+1+\frac{2^n-2}{2^{n-1}}\right)\frac{|0\rangle-|1\rangle}{2^{\frac{(n+1)}{2}}}$$

This means:

$$G_{h=1}|0..01\rangle=\left[\left(-1+\frac{2^n-2}{2^{n-1}}\right)|0..0\rangle+\left(-1+\frac{2^n-2}{2^{n-1}}\right)|0..1\rangle+..+\right.$$
$$\left.+\left(+1+\frac{2^n-2}{2^{n-1}}\right)|\underline{x}\rangle+..+\left(-1+\frac{2^n-2}{2^{n-1}}\right)|1..1\rangle\right]\otimes\frac{|0\rangle-|1\rangle}{2^{\frac{(n+1)}{2}}}$$

which can be written as a block vector:

| | $G_{h=1}\|0..01>$ |
|---|---|
| $\|0..0>$ | $[-1+(2^n-2)/2^{n-1}]\ /2^{n/2}H\|1>$ |
| ... | ... |
| $\|\underline{x}>$ | $[+1+(2^n-2)/2^{n-1}]\ /2^{n/2}H\|1>$ |
| ... | ... |
| $\|1..1>$ | $[-1+(2^n-2)/2^{n-1}]\ /2^{n/2}H\|1>$ |

Now, let us suppose of applying the operator $(D_n\otimes I)\cdot U_F$ to a vector in this form:

| | $\|\varphi>$ |
|---|---|
| $\|0..0>$ | $\alpha H\|1>$ |
| ... | ... |
| $\|\underline{x}>$ | $\beta H\|1>$ |
| ... | ... |
| $\|1..1>$ | $\alpha H\|1>$ |

where $\alpha$ and $\beta$ are real number such that $(2^n\text{-}1)\alpha^2+\beta^2=1$. The result is:

| | $U_F\cdot\|\varphi>$ |
|---|---|
| $\|0..0>$ | $\alpha H\|1>$ |
| ... | ... |
| $\|\underline{x}>$ | $\beta CH\|1>$ |
| ... | ... |
| $\|1..1>$ | $\alpha H\|1>$ |

| | $(D_n\otimes I)\cdot U_F\cdot\|\varphi>$ |
|---|---|
| $\|0..0>$ | $(-\alpha+\Sigma_{j\in\{0,1\}^n-\{\underline{x}\}}\alpha/2^{n-1}-\beta/2^{n-1})H\|1>$ |
| ... | ... |
| $\|\underline{x}>$ | $(+\beta+\Sigma_{j\in\{0,1\}^n-\{\underline{x}\}}\alpha/2^{n-1}-\beta/2^{n-1})H\|1>$ |
| ... | ... |
| $\|1..1>$ | $(-\alpha+\Sigma_{j\in\{0,1\}^n-\{\underline{x}\}}\alpha/2^{n-1}-\beta/2^{n-1})H\|1>$ |

| | $(D_n\otimes I)\cdot U_F\cdot\|\varphi>$ |
|---|---|
| $\|0..0>$ | $\{-\alpha+[(2^n-1)\alpha-\beta]\ /2^{n-1}\}H\|1>$ |
| ... | ... |
| $\|\underline{x}>$ | $\{+\beta+[(2^n-1)\alpha-\beta]\ /2^{n-1}\}H\|1>$ |
| ... | ... |
| $\|1..1>$ | $\{-\alpha+[(2^n-1)\alpha-\beta]\ /2^{n-1}\}H\|1>$ |

This means that if we start from vector $G_{h=1}|0..01>$, which is in the form considered, and we apply $h$ times operator $(D_n\otimes I)\cdot U_F$, the coefficients at time $t$

are such that:

| | |
|---|---|
| $\alpha_t$ | $=2\frac{(2^n-1)\alpha_{t-1}-\beta_{t-1}}{2^n}-\alpha_{t-1}$ |
| $\beta_t$ | $=2\frac{(2^n-1)\alpha_{t-1}-\beta_{t-1}}{2^n}+\beta_{t-1}$ |

So $\beta$ increases while $\alpha$ decreases. Consider for example the vector superposition in Fig. 11.a. By applying the operator ${}^4H$ the vector superposition becomes the superposition of Fig. 11.b. Applying the entanglement operator $U_F$ with $x$=001, the vector superposition of Fig. 11.c is produced and, after the application of $D_n \otimes I$, the superposition is the one depicted in Fig. 11.d. Here, the probability amplitudes of non-interesting vectors are not null, but they are very small.

Suppose to apply operator $U_F$ again: the resulting superposition is reported in Fig. 11.e. Then, by applying $D_n \otimes I$, we obtain the vector linear combination of Fig. 11.f.

We can observe that the probability amplitude of the desired vectors has increased in modulus. This means a greater probability to measure vectors |0010> or |0011>.

If we do measurement after $h$ repetitions of operator $D_n \cdot U_F$, what is the probability $P(h)$ to measure vectors $|x>\otimes|0>$ or $|x>\otimes|1>$? We can show that the:

$$P'(h)=O(2^{-n/2})$$

The quantum block is repeated only 1 time with a sufficiently large $h=O(2^{n/2})$. So, the final collected basis vector is unique.

**Information Analysis**

*A3. Introductory example: Information Analysis of Grover's algorithm*

Let us consider the operator that encoding the input function as:

$$U_F = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}$$

Table 1 represents a general iteration algorithm for information analysis of Grover's QA. In Tables 2 and 3 two iterations of this algorithm are reported. From these tables we observe that:

1. The **entanglement operator** in each iteration increases correlation among the different qubits;

2. The **interference operator** reduces the classical entropy but, as side effect, it destroys part of the quantum correlation measure by the Von Neumann entropy.

Grover algorithm builds in several iterations intelligent states (see Eq.(7)). Every iteration firstly encodes the searched function by entanglement, but then partly destroys the encoded information by the interference operator; several iterations are needed in order to conceal both the need to have encoded information and the need to access it. Grover's algorithm is from Searching Group of Algorithms. The Principle of Minimum Classical (Quantum) Entropy in output of QA means success result on intelligent output states. The searching QA's needed checking minimum of Classical Entropy and co-ordination the gap with Quantum Entropy Amount. The ability of co-ordination of these both values characterizes the intelligence of searching QA's.

## Decoder

As in Deutsch's algorithm, when the output vector from the quantum gate has

been measured, we must interpret it in order to find $\underline{x}$.

From the analyses we did above, this step is very simple. In fact, it is sufficient to choose a large $h$ in order to get the searched vector $|\underline{x}>\otimes|0>$ or $|\underline{x}>\otimes|1>$ with probability near to 1. After getting it we encode back into their binary values the first $n$ basis vector in the resulting tensor product, obtaining string $\underline{x}$ as final answer.

Therefore, it is evinced that Grover's algorithm can be used in search problems.

A search problem can always be so stated: given a truth function $f:\{0,1\}^n \to \{0,1\}$ such that there is only one input $x \in \{0,1\}^n : f(x)=1$ find $x$: this is exactly the kind of problem that can be solved by the Grover's algorithm.

Comparing between them the algorithm above analyzed, it is clear that quantum algorithms have the same structure: a collection of vectors is sequentially submitted to a superposition operator, to an entanglement operator and to an interference operator. The resulting set of vectors is analyzed by a measurement block which extract the desired information.

Finally it must be observed that the differences among quantum algorithms consist essentially in the choice of the interference operator *Int*, the entanglement operator $U_F$ and of the superposition operator $S$.

The input vector is a sort of message that traverses a quantum channel made of three main sub-channels: superposition, entanglement and interference. The entanglement channel is the true input of the algorithm gate. It belongs to a given family depending on the problem to solve and on its input. The superposition and especially the interference channels are chosen in such a way that several measurements effectuated at the end of the channel reveal what kind of entanglement has taken place at the middle of the channel.

In conclusion, it can be stated that Quantum Algorithms are global random

searching algorithms based on the quantum mechanics principles, laws, and quantum effects. In the quantum search, each design variable is represented by a finite linear superposition of classical initial states, with a sequence of elementary unitary steps manipulate the initial quantum state $|i\rangle$ (for the input) such that a measurement of the final state of the system yields the correct output. It begins with elementary classical preprocessing, and then it applies the following quantum experiment: starting in an initial superposition of all possible states, it computes a classical function, applies a Quantum Fast Fourier Transform (*QFFT*), and finally performs a measurement. Depending on the outcome it may carry out one more similar quantum experiments, or complete the computation with some classical post-processing. Usually, three principle operators, i.e. *linear superposition (coherent states), entanglement*, and *interference*, are used in the quantum search algorithm.

In general form the structure of quantum search algorithm can be described as

$$G = \left[ \overbrace{(Int \otimes I_m)}^{\text{Interference}} \circ \underbrace{U_F}_{\text{Entanglement}} \right]^{h+1} \circ \left( \underbrace{H_n \otimes S_m}_{\text{Superposition}} \right) \tag{5}$$

In this quantum algorithms and genetic algorithms structures have the following interrelations:

$$\begin{array}{ll} \text{GA:} & E[m(H,t+1)] \geq m(H,t) \cdot \underbrace{\underbrace{\frac{f(H,t)}{\bar{f}(t)}}_{\text{Selection}} \cdot \underbrace{\{1 - \overbrace{p_c \frac{L(H)}{N-1}(1 - \frac{m(H,t)f(H,t)}{M\bar{f}(t)})}^{P_d(H,t)}\}}_{\text{Crossover}}}_{\Updownarrow} \underbrace{\underbrace{(1-p_m)^{O(H)}}_{\text{Mutation}}}_{\Updownarrow} \\ \text{QA(Gate):} & \left[ \overbrace{(Int \otimes I_m)}^{\text{Interferece}} \circ \overbrace{U_F}^{\text{Entanglemnt}} \right]^{h+1} \circ \left( \underbrace{H_n \otimes S_m}_{\text{Superposition}} \right) \end{array} \tag{6}$$

In Fig. 12 the structures of GA and QSA are compared. In GA a starting population is randomly generated. Mutation and crossover operators are then applied in order to change the genome of some individuals and create some new genomes. Some individuals are then cut off according to a fitness function and

selection of good individuals is done generating a new population. The procedure is then repeated on this new population till an optimum is found.

By analogy in QSA an initial basis vector is transformed into a linear superposition of basis vector by the superposition operator. Quantum operators such as entanglement and interference then act on this superposition of states generating a new one where some states (the non-interesting) have reduced their probability amplitude in modulus and some other (the most interesting) have increased it. The process is repeated several times in order to get to a final probability distribution where an optimum can be easily observed.

The quantum entanglement operator acts in analogy to the genetic mutation operator: in fact it maps every basis vector in the entering superposition into another basis vector by flipping some bits in the ket label. The quantum interference operator acts like the genetic crossover operator by building a new superposition of basis states from the interaction of the probability amplitudes of the states in the entering superposition. But the interference operator includes also the selection operator. In fact, interference increases the probability amplitude modulus of some basis states and decreases the probability amplitude modulus of some other ones according to a general principle, that is maximizing the quantity

$$\Im_T(|output\rangle) = 1 - \frac{E_T^{Sh}(|output\rangle) - E_T^{VN}(|output\rangle)}{|T|} \tag{7}$$

with $T = \{1,\ldots,n\}$. This quantity is called the *intelligence* of the output state and measure how the information encoded into quantum correlation by entanglement is accessible by measurement. The role of the interference operator is in fact to preserve the Von Neumann entropy of the entering entangled state and to reduce at its minimum the Shannon entropy, which has been increased to its maximum by the superposition operator. Note that there is a strong difference between GA and QSA: in GA the fitness functions changes with different instances of the same problem, whereas mutation and crossover are always random. In QSA the fitness

function is always the same (the intelligence of the output state), whereas the entanglement operator strongly depends on the input function $f$.

We propose in this patent to merge the two schemes of GA and QSA undergoing the analogy between them and integrating their peculiarities. The new scheme concerns *quantum genetic search algorithms* (QGSA) and is depicted in Fig. 13.

An initial superposition with $t$ random non-null probability amplitude values is firstly generated

$$|input\rangle = \sum_{i=1}^{t} c_i |x_i\rangle \tag{8}$$

Every ket corresponds to an individual of the population and in the general case is labeled by a real number. So, every individual corresponds to a real number $x_i$ and is implicitly weighted by a probability amplitude value $c_i$. The action of the entanglement and interference operators is genetically simulated: $k$ different paths are randomly chosen, where each path corresponds to the application of an entanglement and interference operator.

The entanglement operator consists into an injective map transforming each basis vector into another basis vector. This is done defining a mutation ray $\varepsilon > 0$ and extracting $t$ different values $\varepsilon_1, \dots, \varepsilon_t$ such that $-\varepsilon \le \varepsilon_i \le \varepsilon$. Then the entanglement operator $\mathbf{U}_F^j$ for path $j$ is defined by the following transformation rule:

$$|x_i\rangle \xrightarrow{\mathbf{U}_F^j} |x_i + \varepsilon_i\rangle \tag{9}$$

When $\mathbf{U}_F^j$ acts on the initial linear superposition, all basis vectors in it undergo mutation

$$\left|\psi^{j}\right\rangle = \sum_{i=1}^{l} c_{i,j}\left|x_i + \varepsilon_{i,j}\right\rangle \tag{10}$$

The mutation operator $\varepsilon$ can be described as following relation

$$\varepsilon = \begin{cases} 1 & for \quad bit \quad permutation \quad 0 \\ 0 & for \quad bit \quad permutation \quad 1 \\ -1 & for \quad phase \quad permutaion \end{cases} \tag{11}$$

Suppose there are eight states in the system, encoded in binary as 000, 001, 010, 011, 100, 110, 111. One of possible states that may be in during a computation is $\frac{i}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}}|100\rangle + \frac{1}{\sqrt{2}}|110\rangle$. A unitary transform is usually constructed so that it is performed at the bit level.

For example, the unitary transformation $\begin{array}{c c} & \begin{array}{cc} 0 & 1 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$ will switch the state $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$ (NOT operator).

Mutation of a chromosome in GA alters one or more genes. It can also be described by changing the bit at a certain position or positions. Switching the bit can be simply carried out by the unitary NOT-transform.

The unitary transformation that acts, as example on the last two bits will transform the state $|1001\rangle$ to state $|1011\rangle$ and the state $|0111\rangle$ to the state $|0101\rangle$ and so on can be described as the following matrix

$$\begin{array}{c c} & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array} \tag{12}$$

is a mutation operator for the set of vectors $|0000\rangle, |0001\rangle, \ldots, |1111\rangle$.

A phase shift operator $Z$ can be described as the following $Z: \begin{array}{ccc} |0\rangle & \rightarrow & |0\rangle \\ |1\rangle & \rightarrow & -|1\rangle \end{array}$ and operator $Y: \begin{array}{ccc} |0\rangle & \rightarrow & |1\rangle \\ |1\rangle & \rightarrow & -|0\rangle \end{array}$ is a combination of negation NOT and a phase shift operator $Z$.

Remark 1. As example, the following matrix

$$\begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ 00 & 1 & 0 & 0 & 0 \\ 01 & 0 & 1 & 0 & 0 \\ 10 & 0 & 0 & 0 & 1 \\ 11 & 0 & 0 & 1 & 0 \end{array} \qquad (13)$$

operates a crossover on the last two bits transforming 1011 and 0110 in 1010 and 0111, where the cutting point is at the middle (one-point crossover).

The two-bit conditional phase shift gate has the following matrix form

$$\begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ 00 & 1 & 0 & 0 & 0 \\ 01 & 0 & 1 & 0 & 0 \\ 10 & 0 & 0 & 1 & 0 \\ 11 & 0 & 0 & 0 & e^{i\phi} \end{array}$$

and the controlled NOT (CNOT) gate that can created entangled states is described by the following matrix:

$$CNOT: \begin{array}{lcl} |00\rangle & \rightarrow & |00\rangle \\ |01\rangle & \rightarrow & |01\rangle \\ |10\rangle & \rightarrow & |11\rangle \\ |11\rangle & \rightarrow & |10\rangle \end{array} \Rightarrow \begin{array}{c c} & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

The interference operator $\mathbf{Int}^1$ is chosen as a random unitary squared matrix of order $t$ whereas the interference operators for the other paths are generated from $\mathbf{Int}^1$ according to a suitable law. Examples of such matrices are the Hadamard transformation matrix $\mathbf{H}_t$ and the diffusion matrix $\mathbf{D}_t$, that have been defined above, but many other ones can be built. The application of entanglement and interference operators produces a new superposition of maximum length $t$:

$$\left|output^j\right\rangle = \sum_{i=1}^{t} c'_{i,j} \left|x_i + \varepsilon_{i,j}\right\rangle \tag{14}$$

The average entropy value for this state is now evaluated. Let $E(x)$ be the entropy value for individual $x$. Then

$$E\left(\left|output^j\right\rangle\right) = \sum_{i=1}^{t} \left\|c'_{i,j}\right\|^2 E\left(x_i + \varepsilon_{i,j}\right) \tag{15}$$

The average entropy value is calculated by averaging every entropy value in the superposition with respect to the squared modulus of the probability amplitudes.

According to this sequence of operations, $k$ different superpositions are generated from the initial one using different entanglement and interference operators. Every time the average entropy value is evaluated. Selection consists in holding only the superposition with minimum average entropy value. When this superposition is obtained, it becomes the new input superposition and the process starts again. The interference operator that has generated the minimum entropy superposition is

hold and $\mathbf{Int}^1$ is set to this operator for the new step. The computation stops when the minimum average entropy value falls under a given critical limit. At this point measurement is simulated, that is a basis value is extracted from the final superposition according to the squared modulus of its probability amplitude.

The whole algorithm is resumed on Fig. 14 in the followings:

1. $|input\rangle = \sum_{i=1}^{t} c_i |x_i\rangle$ with $x_i$ random real numbers and $c_i$ random complex numbers such that $\sum_{i=1}^{t} \|c_i\|^2 = 1$; $\mathbf{Int}^1$ unitary operator of order $t$ randomly generated;

2. $\overline{A} = \begin{pmatrix} \sum_{i=1}^{t} c_i |x_i + \varepsilon_{i,1}\rangle \\ \sum_{i=1}^{t} c_i |x_i + \varepsilon_{i,2}\rangle \\ \cdots \\ \sum_{i=1}^{t} c_i |x_i + \varepsilon_{i,k}\rangle \end{pmatrix}$ with $-\varepsilon \le \varepsilon_{i,j} \le \varepsilon$ randomly generated and $\forall i_1, i_2, j : x_{i_1} + \varepsilon_{i_1,j} \ne x_{i_2} + \varepsilon_{i_2,j}$;

3. $\overline{B} = \begin{pmatrix} \mathbf{Int}^1 \sum_{i=1}^{t} c_i |x_i + \varepsilon_{i,1}\rangle \\ \mathbf{Int}^2 \sum_{i=1}^{t} c_i |x_i + \varepsilon_{i,2}\rangle \\ \cdots \\ \mathbf{Int}^k \sum_{i=1}^{t} c_i |x_i + \varepsilon_{i,k}\rangle \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{t} c'_{i,1} |x_i + \varepsilon_{i,1}\rangle \\ \sum_{i=1}^{t} c'_{i,2} |x_i + \varepsilon_{i,2}\rangle \\ \cdots \\ \sum_{i=1}^{t} c'_{i,k} |x_i + \varepsilon_{i,k}\rangle \end{pmatrix}$ with $\mathbf{Int}^j$ unitary squared matrix of order $t$;

4. $|output^*\rangle = \sum_{i=1}^{t} c'_{i,j^*} |x_i + \varepsilon_{i,j^*}\rangle$ with $j^* = \arg\left(\min\left\{\sum_{i=1}^{t} \|c'_{i,j}\|^2 E(x_i + \varepsilon_{i,j})\right\}\right)$;

5. $\overline{E}^* = \sum_{i=1}^{l} \left\| c'_{i,j^*} \right\|^2 E\left(x_i + \varepsilon_{i,j^*}\right)$

6. If $\overline{E}^* < E^l$ and information risk increment is lower than a pre-established quantity $\Delta$ then extract $x_{i^*} + \varepsilon_{i^*,j^*}$ from the distribution $\left\{x_i + \varepsilon_{i,j^*}, \left\|c'_{i,j^*}\right\|^2\right\}$;

7. Else set $|input\rangle$ to $|output^*\rangle$, $\mathbf{Int}^l$ to $\mathbf{Int}^{l^*}$ and go back to step 2.

*Remark 2.* Step 6 includes methods of accuracy estimation and reliability measurements of the successful result.

The simulation of quantum search algorithm is represented through information flow analysis, information risk increments and entropy level estimations:

1) Applying a quantum gate $G$ on the input vector stores information into the system state, minimizing the gap between the classical Shannon entropy and the quantum von Neumann entropy;

2) Repeating said of applying for calculation (estimation) of information risk increments (see below, *Remark 3*);

3) Measuring said basis vector for estimation the level of average entropy value;

4) Decoding said basis vector of successful result for computation time stopping when the minimum average entropy value falls under a given critical level limit.

*Remark 3.* The information risk increments are calculated (estimated) according to the following formula:

$$-\sqrt{r\left(W^2\right)2I\left(\widetilde{p} : p\right)} \leq (\delta r = \widetilde{r} - r) \leq \sqrt{\widetilde{r}\left(W^2\right)2I\left(p : \widetilde{p}\right)}$$

where:

- $W$ is the loss function;

- $r(W^2) = \iint W^2 p(x,\theta) dx d\theta$ is an average risk for the corresponding probability density function $p(x,\theta)$;
- $x = (x_1, \ldots x_n)$ is a vector of measured values;
- $\theta$ is an unknown parameter;
- $I(p : \tilde{p}) = \iint p(x,\theta) \ln \frac{p(x,\theta)}{\tilde{p}(x,\theta)} dx d\theta$ is the relative entropy (the Kullback-Leibler measure of information divergence).

As stated above, a GA searches for a global optimum in a single solution space. In order to clearly understand the meaning of this statement, further clarifications follows.

A detailed description of the structures of GA and QSA algorithms is showed in Fig. 15. In the GA search, a solution space 301 leads to an initial position (input) 302. The initial position 302 is coded into binary strings using a binary coding scheme 310. GA operators as selection 303, crossover 304, and mutation 305 are applied to the coded strings to generate a population. Through a fitness function 306 (such as a fitness function based on minimum entropy production rate or some other desirable property) a global optimum for the single space 301 is found.

*Example.* The "single solution space" includes any possible coefficient gains of PID controller of plant under stochastic disturbance with fixed statistical properties as the correlation function and probability density function. After stochastic simulation of dynamic behaviour of plant under stochastic excitation with GA we can received the optimal coefficient gains of intelligent PID controller only for stochastic excitation with fixed statistical characteristics. In this case we define the "single space of possible solutions" as 301. If we use stochastic excitation on plant with another statistical characteristics then intelligent PID controller do not can realize control law with fixed KB. In this case we define a new space of possible solutions as 350.

*Remark 4.* If we want the universal look-up table for intelligent PID controller

from many single solution spaces, then the application of GA do not give us a final corrected result (GA - operators not include superposition and quantum correlation as entanglement). GA give the global optimum on the single solution space. In this case we loss the important information about statistical correlation between coefficient gains in the universal look-up table.

By contrast, in the QSA shown in Fig. 15, a group of $N$ solution spaces 350 are used to create an initial position (input) 351. Quantum operators such as superposition 352, entanglement 353, and interference 354 operate on the initial position to produce a measurement. Superposition is created using a Hadamard transformation 361 (one-bit operation). Entanglement is created through a Controlled-NOT (CNOT) operation 362 (a two-bit operation). Interference is created through a Quantum Fourier Transform (QFT) 363. Using the quantum operators, a universal optimum for covering all the spaces in the group 350 is found.

*Remark 5.* Thus, the classical process of selection is loosely analogous to the quantum process of creating a superposition. The classical process of crossover is loosely analogous to the quantum process of entanglement. The classical process of mutation is loosely analogous to the quantum process of interference.

Fig. 16 shows a general structure of a QSA having a conceptual level 400, a structure level 401, a hardware level 402, and software level 403.

At the *conceptual level* 400, an initial state 410 is provided to a process block 420 that creates a *superposition* of states. The superposition of states is provided to a process block 430 that provides a unitary operator $U_f$ to the *entanglement*. An output of the process block 430 is provided to a solution block 440 that computes an *interference* of solutions. An output of the solution block 440 is provided to observation/measurement block 460.

At the *structure level*, an input is coded as a series of quantum bits (qubits) are prepared in an initial state (e.g., a logical zero state) and provided that are

provided to a Hadamard Transform Matrix 421 to produce the superposition. The superposition from the matrix 421 to provide to the operator $U_f$ that produce the entanglement where $U_f$ is in general case a solution to the Schrödinger equation in a process block 431. An output from the process block 431 is provided to a Quantum Fourier Transform (QFT) to provide the interference. The output of the QFT 441 is provided to a Transform Matrix 451. An output of the Transform Matrix 451 is provided as a solution of quantum searching process with maximum probability amplitude 461.

At the *hardware level*, the superposition 420 is produced by rotation gates 422, the operator $U_f$ is implemented as a consequence of elementary gate operations and CNOT-gates 432, the QFFT 441 is implemented as a consequence of Hadamard and Permutation (P) operator gates, and the Transform Matrix 451 is implemented using rotation gates 452.

Fig. 17 illustrates the QSA as an architecture involving the sequence from an initial state, through the creation of superposition. Entanglement is applied to the superposition using quantum parallelism inherent a coherent quantum system with entangled states. The parallelism collapsed when interference is introduced to produce a superposition of solutions through the QFFT. Fig. 17 illustrates these processes by comparing the classical double slit experiment to logical quantum operations and to the quantum search operations.

<u>Remark 6</u>. In the classical double slit, a source creates a particle having an initial superposition of states. This is analogous to the quantum algorithm operation of applying a Hadamard (rotation gates) transform to a qubit initial to an eigenstate. Returning to the double slit, entanglement is produced with the particle passes through the slits. This corresponds to the process of operating on the superposition using the unitary operator $U_f$. Again returning to the double slit, interference is produced when the entangled particles strike a photographic film placed behind the slits to produce an interference pattern (a superposition of solutions). This

corresponds to the QFFT. Finally, selection of the desired solution corresponds to choosing the largest probability from the QFFT (that is, the brightest line produced on the film).

The use of the QSA in connection with a GA or FNN is shown Fig. 18. A generator of initial states 604 works in connection with the GA 605 and, optionally a fuzzy neural network (FNN) 603, to produce a set of initial states. The initial states are provided to a Hadamard transform 602 to produce a superposition of states 601. The superposition of classical states is provided to a process block 606 that introduce entanglement through the use of operators such as a CNOT. An output from the process block 606 is provided to an interference block 607 that computes an interference of the entanglement states using QFFT. An output from the interference block 607 is provided block to a measurement/observation block 608 which selects a desired solution from the superposition of solutions computed by the block 607.

An output from the measurement/observation block 608 is provided to a decision-making block 609. The decision block 609 makes decisions such as inputs for the generator of initial states 604 and, optionally, a new fitness function for the GA 605. The decision block 609 can also provide data to, and receive data from a decoding block 610. The decoding block 610 can communicate with sensors, other control systems, users, etc.

The basis of quantum computation is obtained from the laws of quantum theory wherein information is something that encoded in the state of a physical system, and a computation is something that can be carried out on an actual physical realizable device.

A concrete example showing how a random excitation on control object can "kick" the single space of solutions for fuzzy controller is described in the followings.

*Example.* The KB of the intelligent suspension control system was received with

GA and with stochastic simulation using the random Gaussian signal as road. After on-line simulation with fuzzy controller we use another two real road signal (road measurements in Japan). The results of simulation for pitch angles in Fig. 20 and 21 are shown. Fig. 20 shows that the changing in statistical characteristics of roads (see Fig. 19) kick the single space of solutions for fuzzy controller.

In this case we must repeat the simulation with GA and use *another single space of solution* with the fitness function as entropy production for fuzzy controller with non-Gaussian excitation on control object.

More in details, we are applying GA to minimization of the dynamic behavior of the dynamic system (Plant), and to minimization of entropy production rate. We are using different kinds of random signals (as a stochastic disturbances) which are represents the profiles of roads. Some of this signals were measured from real roads, in Japan, and some of them were created using stochastic simulations with forming filters based on FPK (Fokker – Planck – Kolmogorov) equation. On Fig. 19 three typical road signals are presented. Charts 1901, 1902, 1903 represents changing rates of the signals. Assigned time scale is calculated the way that simulates the vehicle speed of 50kph. First two signals (HouseWC) and (HouseEC) are the measured real roads of Japan. Third signal is a Gaussian road obtained by stochastic simulation with the fixed type of the correlation function. We can see that the dynamic characteristics of this roads are similar (see charts (A)), but statistical characteristics of HouseWC road are very different from statistical characteristics of Gaussian and of HouseEC roads (see charts (B)). HouseWC road represents so called Non-Gaussian (color) stochastic process.

Big difference in the statistical characteristics of the road signals gives us completely different responses from dynamic system and as a result, required different control solutions.

Figs. 20 and 21 illustrate the dynamic and thermodynamic response of the suspension system (PLANT) to the mentioned above excitations. Charts (a) represent dynamic behaviour of the pitch angle of the vehicle under HouseWC

(line1), HouseEC (line 2) and Gaussian (line 3) roads. The knowledge base as a look-up table for fuzzy controller in this case was obtained using Gaussian road signal, and then applied to the HouseWC and HouseEC roads. We see that system response from the roads with the same characteristics are similar, which means that GA has found the optimum solution for Gaussian-like signal shapes, but response obtained from the system with HouseWC road is a completely different signal. For this Non-Gaussian road we need new control GA strategy that differs from the mentioned above e.g. it requires solution from different single space of solutions. On the phase portrait (Charts (b)) the difference of system responses is more visible.

It is desirable, however, to search for a global optimum in multiple solution spaces to find a "universal" optimum. A quantum genetic search algorithm provides the ability to search multiple spaces simultaneously (as described below).

Fig. 2 shows a modified version of the intelligent control system of the invention, wherein a Quantum Genetic Search Algorithm (QGSA) is interposed between the GA and the FNN. The QGSA searches several solution spaces, simultaneously, in order to find a universal optimum, that is, a solution that is optimal considering all solution spaces.

## Accelerator for quantum algorithms

A hardware accelerator for simulating quantum algorithms on a classical computer is described here in below.

The accelerator has a modular structure allowing its generalization to complex models. Starting from known modules, we build an architecture whose objective is the minimization of exponential time required for quantum algorithms simulation on classical computer. The main advantage of this methodology is the possibility of using the logic underlying quantum algorithms in the domain of Genetic Algorithm, opening the way to the new branch of Quantum Genetic

Search Algorithm.

The hardware accelerator is composed by

- *Encoder, Decoder*: these blocks are the real interfaces with classical devices connected to the accelerator.
- *Quantum Block*: it includes all non-classical operations that have to be performed. It is composed of Quantum Gate and Measurement Gate.
  - *Quantum Gate*: it is the core of the accelerator and it is composed three modules that mix the information in a quantum way. The three modules are the following:
    - *Superposition Module*: it depends on the class of problems to solve.
    - *Entanglement Module*: it takes the information from the encoder.
    - *Interference Module*: its operations are iterated until a solution is reached.
  - *Measurement Gate*: extract quantum information through a series of pseudo-random routines.

The final information is sent to the Decoder.

The three blocks of Superposition, Entanglement and Interference, suitably dimensioned for a certain number of qubits, constitute a Quantum Gate, in its general form. These three blocks are then "prepared" in order to implement the desired algorithm. Example of Quantum Gates implementing Grover's Algorithm are depicted in Figs. 23 and 27, and an example of Quantum Gate implementing Deutsch-Jozsa Algorithm is depicted in Fig. 24.

Let us explain more in details how can be designed all the blocks that are enclosed

in the scheme.

- *Superposition*: On this step a tensor product among vectors is needed. This operation can be implemented by means of electric multipliers and mux-demux devices. H-Matrices can be build using ROM cells.
- *Entanglement*: The large matrix $U_F$ , which needs to be modified according to F, can be made using an EPROM device.
- *Interference*: It needs the same kind of blocks needed for Superposition. Only the connection are different.

In the following two examples of quantum gates utilization (see Figs. 24-29) are proposed. Figs. 24 to 26 are relative to Deutsch-Jozsa's Quantum Algorithms for decision making and Figs. 27 to 29 are relative to Grover's Quantum Algorithms for search in unstructured database.

As it can be seen from the above schemes, the core of each subsystem is the Tensor Product Gate that performs the tensor product between a pair of input vectors, outputting a matrix whose components are products of values of all possible different pairs constituted by a component of an input vector and a component of the other vector. A tensor Product Gate can be implemented in a hardware form by suitably connecting a plurality of multipliers.

The quantum gates here represented have only two qubits but they can be improved very easily. In fact each Tensor Product Gate that is added to subsystems provides a doubling of qubits in the gate, if this happens jointly with a suitable re-sizing of mux-demux blocks present in scalar products.

It can also be noted that Interference Blocks in the proposed algorithms seems to be different from the architectural point of view. In fact the simplicity of Deutsch-Jozsa algorithm allows us to adopt a simplified structure, when building a gate implementing only this kind of algorithm. However the structure of Grover Interference is more general and could implement Deutsch-Jozsa, too. This fact

gives to the second example of gate a feature of universality.

For what concerns Quantum Genetic Search Algorithm Structure, its hardware part can be easily derived, with light modifications, from Grover's entanglement and interference blocks. In fact the only difference lies in a random generation of all the matrices, but keeping them unitary in all cases.

An example of a possible embodiment of a random entanglement operator and of a random interference subsystem are depicted in Figs. 30 and 31. The core of the entanglement subsystem is the Dot Product Gate that performs the dot product between a pair of input vectors having the same number of components, outputting a value calculated as the sum of products of paired values of the components of the input vectors. A Dot Product Gate can be implemented in a hardware form by a using at least a multiplier, calculating the products of the paired values, and an adder, summing that products.

**Algorithm for searching in a database**

The Quantum Genetic Search Algorithm of the invention may be satisfactorily used even as an algorithm for searching in a database an item $x_i$ belonging to a set X that is closest to a certain value $Y_0$.

As previously stated, a search problem can always be restated as a Grover's problem. Given that the Quantum Genetic Search Algorithm of the invention may be regarded as a particularly efficient kind of Quantum Algorithm, it is possible to use a Quantum Genetic Search Algorithm for solving a search problem choosing superposition, entanglement and interference operators according to a Grover's algorithm.

More specifically, an algorithm for searching in a database an item $x_i$ belonging to a set X that is closest to a certain value $Y_0$ is defined by:

representing in vector form each item belonging to the set X producing an initial set of vectors;

calculating a second set of vectors by a linear superposition of vectors of the initial set of vectors according to a Grover's Quantum Search Algorithm;

performing on vectors of the second set a certain number of parallel computations of random entanglement operations and random interference operations according to a Grover's Quantum Search Algorithm, producing as many vectors representatives of items of the set X;

associating to each resulting value of the parallel computations a fitness function which must be as greater as smaller the difference between the above mentioned resulting value and the desired value $Y_0$ is;

performing a selection operation of resulting values of the parallel computations according to a Genetic Algorithm using said fitness functions;

identifying the searched item $X_i$ as the final result of the selection.

**Method for controlling a process and relative controller**

The QGSA can be used in a method for controlling a process (PLANT) driven by a control signal (U*). The control signal (U*) is calculated as a function of a parameter adjustment signal (CGS) and an error signal (ε), obtained as the difference between the state of the process (X) and of a reference signal (Y).

The object of the method is to minimize a certain physical quantity, which can be for example the entropy production of the controlled process. To reach this goal, a signal (S) representative of the quantity to be minimized is derived by processing paired values of said state (X) of the process and said control signal (U*).

The method of the invention advantageously uses the Quantum Genetic Search Algorithm to find the best control signal (U*) to input the process to control. Such a control signal is produced by a controller with an adjustable transfer characteristic as a function of a vector of values to assign to the parameters of the controller.

So the problem of controlling a process can be restated in the following manner: find a vector of values that minimizes a certain quantity which is function of said vector, i.e. find a vector that minimizes a certain vector function.

Therefore, it is evident that Genetic Algorithms are of great importance in methods for controlling a process and how a Quantum Genetic Search Algorithm can be effectively used in such applications.

First a correction signal (K2) is periodically calculated from a set of several different values of said control signal (U*), that minimize said derived signal (S) to be minimized.

This correction signal (K2) is calculated applying to a set of vectors representing as many different control signals (U*) a Quantum Genetic Search Algorithm, using a fitness function that is as greater as lower the quantity to be minimized is.

Finally such correction signal (K2) is fed to a fuzzy neural network that produces said parameter adjustment signal (CGS) provided together the error signal (ε) to a fuzzy processor, which adjusts the transfer characteristic of the controller.

Many different embodiments of this method can be implemented by changing the physical quantity to be minimized, which can be for example the difference between the Shannon and von Neumann Entropy or the Heisenberg uncertainty or, more specifically, for an internal combustion engine, the entropy production of the thermodynamic process.

It is also possible to have a suitably adapted Quantum Genetic Search Algorithm of the invention choosing the interference and entanglement operators according to any of quantum problems: for example it is possible to choose such operators according to Grover's problem or a Shor's problem.

A preferred embodiment of the method of the invention consists in running a Genetic Algorithm with the set of vectors representing as many different control signals (U*), producing a temporarily correction signal (K), which is elaborated by

a Quantum Genetic Search Algorithm. This embodiment is preferred because in that way the convergence of the Quantum Genetic Search Algorithm is faster.

A structure of a preferred hardware embodiment of the method of the invention is depicted in Fig. 2. The process (PLANT) driven by a control signal U* is placed in a classical feed-back control loop together with a PID controller. The PID produce the driving signal U* depending on an error signal ε calculated as a function of a state of the process X and of a reference signal Y.

A circuit block derives a signal S representative of a quantity to be minimized, which can be the entropy production for example, calculated by processing paired values of the state X of the process and the control signal U*.

The signal S can be input into a circuit QGSA implementing a Quantum Genetic Search Algorithm, outputting a correction signal K2, or can be first managed by a circuit GA implementing a Genetic Algorithm, that produces a temporary correction signal K to input the QGSA circuit.

Although the circuit implementing the GA is not essential in the structure depicted in Fig. 2, because a Quantum Genetic Search Algorithm of the invention may be considered as a "generalization" of a Genetic Algorithm, generally the circuit GA implementing a Genetic Algorithm is present in the system's architecture.

This because the GA circuit produces structured data for the Quantum Genetic Search Algorithm, allowing a faster convergence of the algorithm. Generally it can be stated that a Genetic Algorithm produces optimal solution from single space of solution: it means that with the GA circuit implementing a Genetic Algorithm we can *compress* information from single space of solution with the guarantee of the safety of *informative* parameters in signal K. The quantum search on structured data guarantees the search of successful solution with higher probability and accuracy than on unstructured data.

A fuzzy neural network FNN produces a driving signal depending on values of

the correction signal K2 output by the QGSA, and a fuzzy controller FC adjusts the transfer characteristic of the classical PID controller depending on values of the driving signal and of the error signal ε.

## Training system for an intelligent control system with reduced number of sensors

The Quantum Genetic Search Algorithm can be used to realize an intelligent control system able to drive a process with a reduced number of sensors as compared to an optimal intelligent control system of the prior art.

Before using a Fuzzy Neural Network (FNN), a "training" phase is necessary in which the FNN learns how to drive the process to be controlled. This "training" is run at the factory or repair center using many different sensors of physical quantities that characterize the running of the process. Generally, in this phase, several sensors that cannot be supported in the normal operation of the process are used. For this reason, it is also necessary to teach the FNN how to drive the process with a reduced number of sensors, i.e. only with these sensors that will be present in the normal operation of the process.

This goal can be reached by the architecture depicted in Fig. 32 and in Fig. 33 in a greater detail. Table 4 is a legend of functional blocks and signals present in the mentioned figures.

The general structure of reduced control system is shown in Figs. 32 and 33. Fig 32 is a block diagram showing a reduced control system 480 and a optimal control system 420. The optimal control system 420, together with an optimizer 440 and a compensator of sensor information 460 are used to teach the reduced control system 480. In Fig. 32, a desired signal (representing the desired output) is provided to an input of the optimal control system 420 and to an input of the reduced control system 480. The optimal control system 420 having $m$ sensors, provides an output sensor signal $x_b$ and an optimal control signal $x_a$. A reduced control system 480 provides an output sensor signal $y_b$ and a reduced control

signal $y_a$. The signals $x_b$ and $y_b$ includes data from $k$ sensors where $k \le m-n$. The $k$ sensors typically being the sensors that are not common between sensor systems 422 and 482. The signals $x_b$ and $y_b$ are provided to first and second inputs of subtractor 491. An output of the subtractor 491 is a signal $\varepsilon_b$, where $\varepsilon_b = x_b - y_b$. The signal $\varepsilon_b$ is provided to a sensor input of a sensor compensator 460. The signals $x_a$ and $y_a$ are provided to first and second inputs of subtractor 490. An output of the subtractor 490 is a signal $\varepsilon_a$ where $\varepsilon_a = x_a - y_a$. The signal $\varepsilon_a$ is provided to a control signal input of the sensor compensator 460. A control information output of sensor compensator 460 is provided to a control information input of the optimizer 440. A sensor information output of the sensor compensator 460 is provided to a sensor information input of the optimizer 440. A sensor signal 483 from the reduced control system 480 is also provided to an input of the optimizer 440. An output of the optimizer 440 provides a teaching signal 443 to an input of the reduced control system 480.

In the description that follows, off-line mode typically refers to a calibration mode, wherein the control object 428 (and the control object 488) is run with an optimal set of $m$ sensors. In one embodiment, the off-line mode is run at the factory or repair center where the additional sensors (i.e., the sensors that are in the $m$ set but not in the $n$ set) are used to train the FNN1 426 and the FNN2 486. The on-line mode typically refers to an operational mode (e.g., normal mode) where system is run with only the $n$ set of sensors.

Fig. 33 is a block diagram, showing the details of the blocks in Fig. 32. In Fig. 33, the output signal $x_b$ is provided by an output of sensor set $m$ 422 having $m$ sensors given by $m = k + n$. The information from the sensor system $m$ 422 is a signal (or group of signals) having optimal information content $I_1$. In other words, the information $I_1$ is the information from the complete set of $m$ sensors in the sensor system 422. The output signal $x_a$ is provided by an output of a control

unit 425. The control signal $x_a$ is provided to an input of a control object 428. An output of the control object 428 is provided to an input of the sensor system 422. Information $I_k$ from the set $k$ of sensors is provided to an on-line learning input of a fuzzy neural network (FNN1) 426 and to an input of a first genetic algorithm (GA1) 427. Information $I_n$ from the set of sensors $n$ in the sensor system 422 is provided to an input of a control object model 424. An off-line tuning signal output from the algorithm GA1 427 is provided to an off-line tuning signal input of the FNN1 426. A control output from the FNN 426 is the control signal $x_a$, which is provided to a control input of the control object 428. The control object model 424 and the FNN 426 together comprise an optimal fuzzy control unit 425.

Also in Fig. 33, the sensor compensator 460 includes a multiplier 462, multiplier 466, an information calculator 464, and an information calculator 468. The multiplier 462 and the information calculator 464 are used in on-line (normal) mode. The multiplier 466 and the information calculator 468 are provided for off-line checking.

The signal $\varepsilon_a$ from the output of the adder 490 is provided to a first input of the multiplier 462 and to a second input of the multiplier 462. An output of the multiplier 462, being a signal $\varepsilon_a^2$, is provided to an input of the information calculator 464. The information calculator 464 computes $H_a(y) \le I(x_a, y_a)$. An output of information calculator 464 is an information criteria for accuracy and reliability $I(x_a, y_a) \to \max$ of control signal in reduced control system.

The signal $\varepsilon_b$ from the output of the adder 491 is provided to a first input of the multiplier 466 and to a second input of the multiplier 466. An output of the multiplier 466, being a signal $\varepsilon_b^2$, is provided to an input of the information calculator 468. The information calculator 468 computes $H_b(y) \le I(x_b, y_b)$. An output of information calculator 468 is an information criteria for accuracy and reliability, $I(x_b, y_b) \to \max$ of output signal of control object with reduced

number of sensors.

The optimizer 440 includes a second genetic algorithm (GA2) 444 and an entropy model 442. The signal $I(x_a, y_a) \to \max$ from the information calculator 464 is provided to a first input of the algorithm (GA2) 444 in the optimizer 440. An entropy signal $S \to \min$ is provided from an output of the thermodynamic model 442 to a second input of the genetic algorithm GA2 444. The signal $I(x_b, y_b) \to \max$ from the information calculator 468 is provided to a third input of the algorithm (GA2) 444 in the optimizer 440.

The signals $I(x_a, y_a) \to \max$ and $I(x_b, y_b) \to \max$ provided to the first and third inputs of the algorithm (GA2) 444 are information criteria, and the entropy signal $S(k_2) \to \min$ provided to the second input of the algorithm (GA2) 444 is a physical criteria based on entropy. An output of the algorithm GA2 444 is a teaching signal for a FNN2 486 described below.

The reduced control system 480 includes a reduced sensor set 482, a control object model 484, the FNN2 486, and control object 488. When run in a special off-line checking (verification) mode, the sensor system 482 also includes the set of sensors $k$. The control object model 484 and the FNN2 486 together comprise a reduced fuzzy control unit 485. An output of the control object 488 is provided to an input of the sensor set 482. An $I_2$ output of the sensor set 482 contains information from the set $n$ of sensors, where $n = (k_1 + k_2) < m$, such that $I_2 < I_1$. The Information $I_2$ is provided to a tuning input of the FNN2 486, to an input of the control object model 484 and to an input of the entropy model 442. The teaching signal 443 from the algorithm GA2 444 is provided to a teaching signal input of the FNN2 486. A control output from the FNN2 486 is the signal $y_a$, which is provided to a control input of the control object 488.

The control object models 424 and 484 may be either full or partial models. A full mathematical model representation of control objects is a description with

differential equations including dissipative processes, while a partial model is a model obtained without a complete analytical description.

As example, for a suspension control system, it is possible to write non-linear equations for the system "car + suspension" and calculate in analytical form the entropy production rate using dissipative terms of nonlinear equations, while for an engine control system the analytical description of mathematical model is not available.

Although Figs. 32 and 33 show the optimal system 420 and the reduced system 480 as separate systems, typically the system 420 and 480 are the same system. The system 480 is "created" from the system 420 be removing the extra sensors and training the neural network. Thus, typically, the control object models 424 and 484 are the same. The control object 428 and 488 are also typically the same.

Fig. 33 shows arrow an off-line tuning arrow mark from GA1 427 to the FNN1 426 and from the GA2 444 to the FNN2 486. Fig. 33 also shows an on-line learning arrow mark 429 from the sensor system 422 to the FNN1 426. The tuning of the GA2 444 means to change a set of bonding coefficients in the FNN2 486. The bonding coefficients are changed (using, for example, an iterative backward propagation or trial error process) so that $I(x,y)$ trends toward a maximum and $S$ trends toward a minimum. In other words, the information of the coded set of the coefficients is sent to the FNN2 486 from GA2 444 as $I(x,y)$ and $S$ are evaluated. Typically, the bonding coefficients in the FNN2 486 are tuned in off-line mode at a factory or service center.

The teaching signal 429 is a signal that operates on the FNN1 426 during operation of the optimal control system 420 with an optimal control set. Typically, the teaching signal 429 is provided by sensors that are not used with the reduced control system 480 when the reduced control system is operating in on-line mode. The GA1 427 tunes the FNN1 426 during off-line mode. The signal lines associated with $x_b$ and $y_b$ are dashed to indicate that the $x_b$ and $y_b$ signals are

typically used only during a special off-line checking (i.e., verification) mode. During the verification mode, the reduced control system is run with an optimal set of sensors. The additional sensor information is provided to the optimizer 440 and the optimizer 440 verifies that the reduced control system 480 operates with the desired, almost optimal, accuracy.

For stable and unstable control objects with a non-linear dissipative mathematical model description and reduced number of sensors (or a different set of sensors) the control system design is connected with calculation of an output accuracy of the control object and reliability of the control system according to the information criteria $I(x_a, y_a) \to \max$ and $I(x_b, y_b) \to \max$. Control system design is also connected with the checking of stability and robustness of the control system and the control object according to the physical criteria $S(k_2) \to \min$.

**In a first step**, the genetic algorithm GA2 444 with a fitness function described as the *maximum of mutual information* between an optimal control signals $x_a$ and a reduced control signal $y_a$ is used to develop teaching signal 443 for the fuzzy neural network FNN2 486 in an off-line simulation. The fuzzy neural network FNN2 486 is realized using a learning process with back-propagation of error for adaptation to the teaching signal, and to develop a look-up table for changing the parameters of PID-controller in the controller 485. This provides sufficient conditions for achieving the desired control reliability with sufficient accuracy.

**In a second step**, the genetic algorithm GA2 444, with a fitness function described as the minimum entropy, $S$ or production rate $dS/dt$ , (calculated according to a mathematical model or experimental results of sensor measurement information of the control object 488), is used to realize a node-correction look-up table in the FNN2 486. This approach provides stability and robustness of the reduced control system 480 with reliable and sufficient accuracy of control. This provides a sufficient condition of design for the robust intelligent control system

with the reduced number of sensors.

The first and second steps above need not be done in the listed order or sequentially. In the simulation of unstable objects, both steps are preferably done in parallel using the fitness function as the sum of the physical and the information criteria.

After the simulation of a look-up table for the FNN2 486, the mathematical model of control object 484 is changed on the sensor system 482 for checking the qualitative characteristics between the reduced control system with the reduced number of sensors and the reduced control system with a full complement (optimum number) of sensors. Parallel optimization in GA2 444 with two fitness functions is used to realize the global correction of a look-up table in the FNN2 486.

The entropy model 442 extracts data from the sensor system information $I_2$ to help determine the desired number of sensors for measurement and control of the control object 488.

Figs. 32 and 33 show the general case when the reduction of sensors excludes the measurement sensors in the output of the control object and the calculation comparison of control signal of information criteria is possible. The sensor compensator 460 computes the information criteria as a maximum of mutual information between the two control signals $x_a$ and $y_a$ (used as the first fitness function for the GA2 444). The entropy model 442 present the physical criteria as a minimum production entropy (used as the second fitness function in GA2 444) using the information from sensors 482. The output of GA2 444 is the teaching signal 443 for the FNN2 486 that is used on-line to develop the reduced control signal $y_a$ such that the quality of the reduced control signal $y_b$ is similar to the quality of the optimal control signal $x_a$. Thus, the optimizer 440 provides stability and robustness of control (using the physical criteria), and reliability with sufficient accuracy (using the information criteria).

With off-line checking, the optimizer 440 provides correction of the control signal $y_a$ from the FNN2 486 using new information criteria. Since the information measure is additive, it is possible to do the on-line/off-line steps in sequence, or in parallel. In off-line checking, the sensor system 482 typically uses all of the sensors only for checking of the quality and correction of the control signal $y_a$. This approach provides the desired reliability and quality of control, even when the control object 488 is unstable.

For the reduced sensor system 482 (with $n$ sensors) the FNN2 486 preferably uses a learning and adaptation process instead of a Fuzzy Controller (FC) algorithm.

If the Control Object will work in different environments, with different statistical characteristics, then, global optimizer 450 is used. Global optimizator 450 includes GA2 444 and QGSA 448. Output 449 of GA2 is input of QGSA 448. Output of QGSA 448 is a teaching signal for FNN2 486. Output 449 must be generated for each single solution space.

**Reduced control system applied to an internal combustion engine**

In one embodiment, the reduced control system is applied to an ***internal combustion*** piston engine, a jet engine, a gas-turbine engine, a rocket engine etc., to provide control without the use of extra sensors, such as, for example, an oxygen sensor.

Fig. 34 shows an internal combustion piston engine, having four sensors, an intake air temperature sensor 602, a water temperature sensor 604, a crank angle sensor 608. The air temperature sensor 602 measures the air temperature in an intake manifold 620. A fuel injector 629 provides fuel to the air in the intake manifold 620. The intake manifold 620 provides air and fuel to a combustion chamber 622. Burning of the fuel-air mixture in the combustion chamber 622 drives a piston 628. The piston 628 is connected to a crank 626 such that movement of the piston 628 turns the crank 626. The crank angle sensor 606

measures the rotational position of the crank 626. The water sensor measures the temperature of the water in a water jacket 630 surrounding the combustion chamber 622 and the piston 628. Exhaust gases from the combustion chamber 622 are provided to an exhaust manifold 624 and the air-fuel ratio sensor 608 measures the ratio of air to fuel in the exhaust gases.

Fig. 35 is a block diagram showing a reduced control system 780 and an optimal control system 720. The optimal control system 720, together with an optimizer 740 and a sensor compensator 760 are used to teach the reduced control system 780. In Fig. 35, a desired signal (representing the desired engine output) is provided to an input of the optimal control system 720 and to an input of the reduced control system 780. The optimal control system 720, having five sensors, provides an optimal control signal $x_a$ and an sensor output signal $x_b$. A reduced control system 780 provides a reduced control output signal $y_a$, and an output sensor signal $y_b$. The signals $x_b$ and $y_b$ include data from the A/F sensor 608. The signals $x_b$ and $y_b$ are provided to first and second inputs of a subtractor 791. An output of a subtractor 791 is a signal $\varepsilon_b$ where $\varepsilon_b = x_b - y_b$. The signal $\varepsilon_b$ is provided to a sensor input of a sensor compensator 760. The signals $x_a$ and $y_a$ are provided to first and second inputs of a subtractor 790. An output of the subtractor 790 is a signal $\varepsilon_a$ where $\varepsilon_a = x_a - y_a$. The signal $\varepsilon_a$ is provided to a control signal input of the sensor compensator 760. A control information output of the sensor compensator 760 is provided to a control information input of the optimizer 740.A sensor information output of the sensor compensator 760 is provided to a sensor information input of the optimizer 740. A sensor signal 783 from the reduced control system 780 is also provided to an input of the optimizer 740.An output of the optimizer 740 provides a teaching signal 747 to an input of the reduced control system 780.

The output signal $x_b$ is provided by an output of a sensor system 722 having five sensors, including the intake temperature sensor 602, the water temperature sensor

604, the crank angle sensor 607, and air-fuel ration (A/F) sensor 608. The information from the sensor system 722 is a group of signals having optimal information content $I_1$. In other words, the information $I_1$ is the information from the complete set of five sensors in the sensor system 722.

The output signal $x_a$ is provided by an output of a control unit 725. The control signal $x_a$ is provided to an input of an engine 728. An output of engine 728 is provided to an input of the sensor system 722. Information $I_{k1}$ from the A/F sensor 608 is provided to an on-line learning input of a fuzzy neural network (FNN) 726 and to an input of a first Genetic Algorithm (GA1) 727. Information $I_{k1}$ from the set of four sensors excluding the A/F sensor 608 is provided to an input of an engine model 724. An off-line tuning signal output from the algorithm GA1 727 is provided to an off-line tuning signal input of the FNN 726. A control output from the FNN 726 is a fuel injector the control signal $U_1$, which is provided to a control input of the engine 728. The signal $U_1$ is also the signal $x_a$. The engine model 724 and the FNN 726 together comprise an optimal control unit 725.

The sensor compensator 760 includes a multiplier 762, a multiplier 766, an information calculator 764. The multiplier 762 and the information calculator 764 are used in on-line (normal) mode. The multiplier 766 and the information calculator 768 are provided for off-line checking.

The signal $\varepsilon_a$ from the output of the adder 790 is provided to a first input of the multiplier 762 and to a second input of the multiplier 762. An output of the multiplier 762, being a signal $\varepsilon_a^2$, is provided to an input of the information calculator 764. The information calculator 764 computes $H_a(y) \le I(x_a, y_a)$. An output of the information calculator 764 is an information criteria for accuracy and reliability, $I(x_a, y_a) \rightarrow \max$.

The signal $\varepsilon_b$ from the output of the adder 791 is provided to a first input of the multiplier 766 and to a second input of the multiplier 766. An output of the multiplier 764, being a signal $\varepsilon_a^2$, is provided to an input of the information calculator 768. The information calculator 768 computes $H_b(y) \le I(x_b, y_b)$. An output of the information calculator 768 is an information criteria for accuracy and reliability, $I(x_b, y_b) \to \max$.

The optimizer 740 includes a second Genetic Algorithm (GA2) 744 and a thermodynamic (entropy) model 742. The signal $I(x_a, y_a) \to \max$ from the information calculator 764 is provided to a first input of the algorithm (GA2) 744 in the optimizer 740. An entropy signal $S \to \min$ is provided form an output of the thermodynamic model 742 to a second input of the algorithm (GA2) 744. The signal $I(x_b, y_b) \to \max$ from the information calculator 768 is provided to a third input of the algorithm (GA2) 744 in the optimizer 740.

The signals $I(x_a, y_a) \to \max$ and $I(x_b, y_b) \to \max$ provided to the first and third inputs of the algorithm (GA2) 744 are information criteria, and the entropy signal $S(k2) \to \min$ provided to the second input of the algorithm GA2 744 is a physical criteria based on entropy. An output of the algorithm GA2 744 is a teaching signal for the FNN 786.

The reduced control system 780 includes a reduced sensor system 782, an engine model 784, the FNN 786, and an engine 788. The reduced sensor system 782 includes all of the engine sensors in the sensor system 722 except the A/F sensor 608. When run in a special off-line checking mode, the sensor system 782 also includes A/F sensor 608. The engine model 784 and the FNN 786 together comprise a reduced control unit 785. An output of the engine 788 is provided to an input of the sensor set 782. An $I_2$ output of the sensor set 782 contains information from four engine sensors, such that $I_2 < I_1$. The information $I_2$ is provided to an input of the control object model 784, and to an input of the

thermodynamic model 742. The teaching signal 747 from the algorithm GA2 744 is provided to a teaching signal input of the FNN 786. A control output from the FNN 786 is an injector control signal $U_2$, which is also the signal $y_a$.

Operation of the system shown in Fig. 35 is in many respects similar to the operation of the system shown in Figs. 32 and 33.

The thermodynamic model 742 is built using the thermodynamic relationship between entropy production and temperature information from the water temperature sensor 604 ($T^W$) and the air temperature sensor 602 ($T^A$). The entropy production $S(T^W,T^A)$, is calculated using the relationship

$$S = c\frac{\left[\ln\left(\frac{T^W}{T^A}\right)\right]^2}{\Delta\tau - \ln\left(\frac{T^W}{T^A}\right)} \tag{16}$$

where $\Delta\tau$ is the duration of a finite process.

The external specific work between two arbitrary states follows in the form

$$I(T^i,T^f,\tau^i,\tau^f) = c(T^i - T^f) - cT^e \ln\frac{T^i}{T^f} - cT^e\frac{\left[\ln\frac{T^i}{T^f}\right]^2}{\tau^f - \tau^i - \ln\frac{T^i}{T^f}} \tag{17}$$

From equation (17), with $T^i = T^W$ and $T^f = T^A$, then the minimal integral of the entropy production is

$$S_\sigma = c\frac{\left[\ln\frac{T^W}{T^A}\right]^2}{\Delta\tau - \ln\left(\frac{T^W}{T^A}\right)}, \qquad \text{where } \Delta\tau = \tau^f - \tau^i \tag{18}$$

The function from equation (17) satisfies the backward Hamilton – Jacobi equation.

Fig. 36 is used QGSA in the case when partial engine model presented as a stochastic A/F ratio (Bounded control). It means, that statistical properties of stochastic A/F ratio are modeled with different density probability functions (Gaussian, Non-Gaussian as Uniform distribution, Rayleigh distribution and etc.).

**Reduced control system applied to a vehicle suspension**

Another embodiment consists of a control system for a vehicle suspension to provide control of the suspension system using data from a reduced number of sensors. In this case the optimized control system provides an optimum control signal based on data obtained from a plurality of angle (pitch and roll) and position sensors.

An embodiment of an intelligent control system of a vehicle suspension can be obtained using a similar hardware architecture to the one depicted in Fig. 36 for an internal combustion engine. The main difference consists in substituting several functional blocks with the appropriate functional blocks for a vehicle suspension. For example, blocks 728 and 788 must be the vehicle suspension system, blocks 724 and 784 must be a vehicle suspension model, sensors 722 and 782 must be the appropriate sensors for monitoring a vehicle suspension, such as position sensors and roll angle sensors.

## CLAIMS

1. A method of controlling a process (PLANT) driven by a control signal (U*) for producing a corresponding output, comprising

producing an error signal (ε) as a function of a state of said process (X) and of a reference signal (Y),

generating a control signal (U*) as a function of said error signal (ε) and of a parameter adjustment signal (CGS) and feeding it to said process (PLANT),

deriving a signal (S) representative of a quantity to be minimized calculated by processing paired values of said state (X) of the process and said control signal (U*),

calculating a correction signal (K2) from a set of several different values of said control signal (U*), that minimize said derived signal (S) to be minimized,

calculating said parameter adjustment signal (CGS) by a neural network and fuzzy logic processor from said error signal (ε) and correction signal (K2), characterized in that

said correction signal (K2) is periodically calculated by a Quantum Genetic Search Algorithm consisting in a merging of a genetic algorithm and a quantum search algorithm comprising the following steps:

producing an initial set of vectors representing in vector form several different possible values of said correction signal (K2);

calculating a second set of vectors by a linear superposition of vectors of said initial set;

performing on vectors of said second set a certain number of parallel computations of random entanglement operations and random interference operations according to a quantum search algorithm, producing as many vectors representatives of values of said control signal (U*) of the process (PLANT);

associating to each resulting value of said parallel computations a fitness function as greater as lower the quantity to be minimized (S), calculated using paired values of said control signal (U*) and state (X) of the process (PLANT)

corresponding to the considered resulting value, is;

performing a selection operation of resulting values of said parallel computations according to a genetic algorithm using said fitness functions;

calculating said correction signal (K2) as the final result of said selection.

2. The method according to claim 1 wherein said initial set of vectors is obtained by

producing a first set of vectors representing several different possible values of said correction signal (K2);

applying a genetic algorithm (GA) on said first set of vectors calculating a temporary correction signal (K) using a fitness function as greater as lower the quantity to be minimized (S), calculated using paired values of said control signal (U*) and state (X) of the process (PLANT) corresponding to the considered resulting value, is;

producing said initial set as a collection of the different values of said temporary correction signal (K).

3. The method of claims 1 or 2 wherein said derived signal (S) to be minimized represents the information distance between Shannon entropy and von Neumann entropy of pairs of said control signal (U*) and state (X);

said fitness function is the information intelligence measure of each resulting value of said parallel computations.

4. The method of claims 1 or 2 wherein said derived signal (S) to be minimized is the Heisenberg uncertainty.

5. The method according to one of the preceding claims wherein said quantum search algorithm is a Grover quantum search algorithm.

6. The method according to one of the preceding claims wherein said linear superposition is a Hadamard rotation.

7. The method according to one of the preceding claims wherein said interference

operation is obtained by an interference operator constituted with a random unitary squared matrix implementing a Fast Quantum Fourier Transform.

8. A controller of a process (PLANT) driven by a control signal (U*) comprising

a first circuit block producing an error signal (ε) as a function of a state of said process (X) and of a reference signal (Y),

a second circuit block with adjustable transfer characteristic generating a control signal (U*) as a function of said error signal (ε) and feeding it to said process (PLANT),

a third circuit block deriving a signal (S) representative of a quantity to be minimized calculated by processing paired values of said state (X) of the process and said control signal (U*),

a fourth circuit block calculating a correction signal (K2) from a set of several different values of said control signal (U*), that minimize said derived signal (S) representing the entropy production capacity,

a fuzzy neural network producing a driving signal depending on values of said correction signal (K2),

a fuzzy controller adjusting the transfer characteristic of said second circuit block depending on values of said driving signal and said error signal (ε),

characterized in that said fourth circuit block performs the following operations

producing an initial set of vectors representing in vector form several different possible values of said correction signal (K2);

calculating a second set of vectors by a linear superposition of vectors of said initial set;

performing on vectors of said second set a certain number of parallel computations of random entanglement operations and random interference operations according to a quantum search algorithm, producing as many vectors representatives of values of said control signal (U*) of the process (PLANT);

associating to each resulting value of said parallel computations a fitness function as greater as lower the quantity to be minimized (S), calculated using paired values of said control signal (U*) and state (X) of the process (PLANT) corresponding to the considered resulting value, is;

performing a selection operation of resulting values of said parallel computations according to a genetic algorithm using said fitness functions;

calculating said correction signal (K2) as the final result of said selection.

9. The controller of the preceding claim wherein said initial set of vectors is obtained by

producing a first set of vectors representing several different possible values of said correction signal (K2);

calculating a temporary correction signal (K) by a genetic algorithm (GA) applied on said first set of vectors using a fitness function as greater as lower the quantity to be minimized (S), calculated using paired values of said control signal (U*) and state (X) of the process (PLANT) corresponding to the considered resulting value, is;

producing said initial set as a collection of the different values of said temporary correction signal (K).

10. The controller according to one of claims 8 or 9 wherein said process to control (PLANT) is an internal combustion engine and said third circuit derives a signal (S) representative of the entropy production of said engine.

11. The controller according to one of claims 8 or 9 wherein said derived signal (S) to be minimized represents the information distance between Shannon entropy and von Neumann entropy of pairs of said control signal (U*) and state (X);

said fitness function is the information intelligence measure of each resulting value of said parallel computations.

12. The controller according to one of claims from 8 to 11 wherein said second circuit block is a proportional-integral-differential controller (PID).

13. A method of searching in a database an item ($x_i$) belonging to a set (X) that is closest to a certain value ($Y_0$), consisting in a merging of a genetic algorithm and a quantum search algorithm comprising the following steps:

representing in vector form each item belonging to said set (X) producing an initial set of vectors;

calculating a second set of vectors by a linear superposition of vectors of said initial set of vectors according to a Grover quantum search algorithm;

performing on vectors of said second set a certain number of parallel computations of random entanglement operations and random interference operations according to a Grover quantum search algorithm, producing as many vectors representatives of items of said set (X);

associating to each resulting value of said parallel computations a fitness function as greater as smaller the difference between said resulting value and said desired value ($Y_0$) is;

performing a selection operation of resulting values of said parallel computations according to a genetic algorithm using said fitness functions;

identifying said searched item ($x_i$) as the final result of said selection.

14. A random entanglement gate for carrying out random entanglement operations according to a quantum search algorithm on an input set of vectors comprising

a certain number (N) of random generators each producing a respective random vector,

said certain number (N) of dot gates each composed of at least a multiplier calculating products of paired values of the components of a pair of input vectors having the same number of components, and at least an adder outputting the sum of said products, each coupled with a respective random vector and with a respective vector of said input set of vectors, producing a random entanglement set of vectors.

15. A random interference gate for carrying out random interference operations according to a Grover quantum search algorithm on an input set of vectors

comprising

a circuit producing a random diffusion matrix, constituted by

an identity block outputting an identity matrix,

a random generator outputting a random unitary matrix,

a first tensor gate, composed of at least a multiplier calculating components of an output matrix as products of values of all possible different pairs constituted by a component of a first input vector and a component of a second input vector, coupled with said identity matrix and said random unitary matrix, producing a first matrix (4XN1),

a second tensor gate, similar to said first tensor gate, coupled with said first matrix (4XN1) and with said random unitary matrix, producing a second matrix (4XN2),

a third tensor gate, similar to said first tensor gate, being input with two copies of said identity matrix, producing a third matrix (4XN3),

a fourth tensor gate, similar to said first tensor gate, coupled with said first matrix (4XN3) and with said identity matrix, producing a fourth matrix (4XN4),

a subtractor coupled with said matrices second (4XN2) and fourth (4XN4), producing said random diffusion matrix as the difference between said second matrix (4XN2) and said fourth matrix (4XN4);

a block outputting the result of said interference operation by calculating a scalar product between said random diffusion matrix and said set of input vectors.

16. An accelerator for quantum algorithm execution by a classical computer for finding a certain property of a bit vector function (*f*) being processed, composed of

an encoder producing from said bit vector function (*f*) a set of values representing a transfer matrix ($U_F$);

an accelerator core (QUANTUM_BLOCK) comprising:

a quantum gate implementing in succession a linear superposition, an

entanglement operation and interference operation on an input set of vectors according to a quantum algorithm using said transfer matrix ($U_F$) as entanglement operator, producing a second set of vectors,

a block (MEASUREMENT) calculating for each vector of said second set a probability coefficient as the squared root of the sum of squares of components of said vector respect said input set of vectors, and selecting an output vector from said second set by running a random choice routine using said probability coefficients;

a decoder comparing said output vector in a look-up table and outputting a signal representative of the property of said bit vector function (*f*) corresponding to said output vector.

17. The accelerator of claim 16 wherein said encoder produces said set of values representing said transfer matrix ($U_F$) as a set of values representing the transfer matrix of an injective bit vector function (F) whose domain dimension is equal to the sum between the domain dimension N and of the co-domain dimension M of said function (*f*), transforming each bit vector (A) of its domain in a respective bit vector (B) obtained by linking the first N bits of said vector (A) to a bit vector obtained by carrying out a XOR operation between the last M bits of said vector (A) and the corresponding bits of a vector obtained applying said bit vector function (*f*) to a bit vector constituted by the first N bits of said vector (A).

18. The accelerator according to one of claims 16 or 17 wherein

said quantum gate is composed of a superposition subsystem executing said linear superposition, an entanglement subsystem executing said entanglement operation and an interference subsystem executing said interference operation according to a Deutsch-Jozsa quantum algorithm,

said superposition subsystem being composed of

a set of a number N+1 of matrix blocks, each one being input with a respective vector of said input set of vectors and producing a respective rotated vector obtained by carrying out a Hadamard rotation,

a first tensor gate, composed of at least a multiplier calculating components of an output matrix as products of values of all possible different pairs constituted by a component of a first input vector and a component of a second input vector, carrying out the tensor product of a number N of said rotated vectors, producing a certain matrix (2XN),

a second tensor gate similar to said first tensor gate, carrying out the tensor product of said certain matrix (2XN) and the rotated vector not being input in said first tensor gate, producing a linear superposition set of vectors; said entanglement subsystem producing an entanglement set of vector by applying said transfer matrix ($U_F$) to said linear superposition set of vectors; said interference subsystem being composed of

a tensor gate similar to said first tensor gate, being input with a Hadamard matrix (H) and an identity matrix (I), producing an output matrix (OUT1),

a dot gate, composed of at least a multiplier calculating products of paired values of the components of a pair of input vectors having the same number of components, being input with said output matrix (OUT1) and with the vector produced by said entanglement operator, producing said second set of vectors.

19. The accelerator according to one of claims 16 or 17 wherein

said quantum gate is composed of a superposition subsystem executing said linear superposition, an entanglement subsystem executing said entanglement operation and an interference subsystem executing said interference operation according to a Grover quantum algorithm,

said superposition subsystem being composed of

a set of a number N+1 of matrix blocks, each one being input with a respective vector of said input set of vectors and producing a respective rotated vector obtained by carrying out a Hadamard rotation,

a first tensor gate, composed of at least a multiplier calculating components of an output matrix as products of values of all possible

different pairs constituted by a component of a first input vector and a component of a second input vector, carrying out the tensor product of a number N of said rotated vectors, producing a certain matrix (2XN),

a second tensor gate, similar to said first tensor gate, carrying out the tensor product of said certain matrix (2XN) and the rotated vector not being input in said first tensor gate, producing a linear superposition set of vectors; said entanglement subsystem producing an entanglement set of vectors by applying said transfer matrix ($U_F$) to said linear superposition set of vectors; said interference subsystem being composed of

a dot gate, composed of at least a multiplier calculating products of paired values of the components of a pair of input vectors having the same number of components, being input with said entanglement set of vectors and with a diffusion matrix ($D_N$) producing said second set of vectors.

20. A training system of an intelligent control system with reduced number of sensors comprising

an intelligent control system with a certain number of sensors (m) constituted by

a replica of a process to control (428),

a fuzzy control unit (425) producing a driving signal ($x_a$) being input to said replica process (428),

a first set of a second number (k) of sensors of as many states of said replica process producing a sensor signal ($x_b$) fed to said fuzzy control unit,

a second set of a third number of sensors (k1) of as many states of said replica process producing a second sensor signal (4222) fed to said fuzzy control unit,

a third set of a fourth number of sensors (k2) of as many states of said replica process producing a third sensor signal (4221),

a circuit block implementing a first genetic algorithm (GA1) calculating from said third sensor signal (4221) a teaching signal input to said fuzzy control unit (425);

an intelligent control system to be trained with a fifth number of sensors (n) constituted by

a process to control (488),

a second fuzzy control unit (485) producing a second driving signal ($y_a$) being input to said process to control (488),

a replica of said first set of said second number (k) of sensors of as many states of said process producing a fourth sensor signal ($y_b$),

a replica of said second set of said third number of sensors (k1) sensing as many states of said process producing a fifth sensor signal (4822) fed to said second fuzzy control unit (485),

a replica of said third set of said fourth number of sensors (k2) of as many states of said process (488) producing a sixth sensor signal (4821),

a circuit block producing from said fifth sensor signal (4821) a signal (S[k]) representative of a physical quantity to be minimized (S),

a global optimizer (450) deriving a teaching signal (443) for said second fuzzy controller (485) from said signal (S[k]) and from a pair of signals representing the mutual information between said first driving signal ($x_a$) and said second driving signal ($y_a$) and the mutual information between said sensor signal ($x_b$) and said fourth sensor signal ($y_b$), respectively;

a compensator of sensor information (450) calculating said pair of signals representing the mutual information between said first driving signal ($x_a$) and said second driving signal ($y_a$) and the mutual information between said sensor signal ($x_b$) and said fourth sensor signal ($y_b$), respectively.

21. The training system of claim 20 wherein said global optimizer (450) is a block deriving said teaching signal (443) by implementing a second Genetic Algorithm.

22. The training system of claim 20 wherein said global optimizer (450) is a block deriving said teaching signal (443) by implementing a Quantum Genetic Search Algorithm as described in claim 1.

23. The training system of claims 21 and 22 wherein said global optimizer (450) is a block deriving said teaching signal (443) as the final result of said Quantum Genetic Search Algorithm applied on the output of said second Genetic Algorithm.

24. The training system according to one of claims from 20 to 23 wherein said physical quantity to be minimized is the entropy production of said process (488).

25. The training system of any of the claims from 20 to 24 wherein

said process to control (488) is an internal combustion engine and said replica process (428) is a replica of said internal combustion engine,

said first set of sensors comprises an air fuel ratio sensor,

said second set of sensors comprises a crank angle sensor and a pressure sensor,

said third set of sensors comprises an air temperature sensor and a water temperature sensor.

26. The training system of any of the claims from 20 to 24 wherein

said process to control (488) is a vehicle suspension and said replica process (428) is a replica of said vehicle suspension,

at least one of said sets of sensors comprises at least one sensor belonging to the group composed of position sensors, pitch sensors and roll angle sensors.

27. A method for training a fuzzy control unit (485) of an intelligent control system (480) comprising

a process to control (488),

a fuzzy control unit (485) producing a first driving signal ($y_a$) being input to said process to control (488),

a first set of a first number (k) of sensors of as many states of said process (488) producing a first sensor signal ($y_b$),

a second set of said second number of sensors (k1) sensing as many

states of said process (488) producing a second sensor signal (4822) fed to said fuzzy control unit (485),

a third set of a third number of sensors (k2) of as many states of said process (488) producing a third sensor signal (4821),

a circuit block producing from said third sensor signal (4821) a signal (S[k]) representative of a physical quantity to be minimized (s),

a global optimizer (450) producing a temporary teaching signal (442) for said fuzzy control unit (485),

using a control system (420) of a replica (428) of said process to control comprising

a replica of said process to control (428),

a control unit (425) producing a second driving signal ($x_a$) being input to said replica process (428),

a replica of said sets first, second and third of said numbers of sensors first (k), second (k1) and third (k2) of as many states of said replica process (428), respectively, producing sensor signals fourth ($x_b$), fifth (4222) and a sixth (4221), respectively, fed to said control unit,

characterized in that it comprises the following steps:

inputting both control systems (480, 420) with a signal representative of a desired output provided from outside;

calculating the mutual information (I1) of paired values of said driving signals first ($y_a$) and second ($x_a$);

calculating the mutual information (I2) of paired values of said sensor signals first ($y_b$) and fourth ($x_b$);

inputting said fuzzy control unit (485) with a final teaching signal (443) calculated in function of said temporary teaching signal (442) said mutual information of driving signals (I1) and said mutual information of sensor signals (I2).

28. The method of claim 27 wherein said final teaching signal (443) is derived by implementing a Genetic Algorithm on said temporary teaching signal

(442) using a fitness function depending on said mutual information values (I1, I2).

29. The method of claim 27 wherein said final teaching signal (443) is derived by implementing a Quantum Genetic Search Algorithm, as described in claim 1, on said temporary teaching signal (442) using a fitness function depending on said mutual information values (I1, I2).

30. The method of claim 28 wherein said final teaching signal (443) is derived by implementing a Genetic Algorithm on said temporary teaching signal (442) using a fitness function depending on said mutual information values (I1, I2) and by applying to a value calculated by said Genetic Algorithm a Quantum Genetic Search Algorithm.

31. The method according to one of claims from 27 to 30 wherein said physical quantity to be minimized (S) is the entropy production of said process to control (488).

32. The method according to one of claims from 27 to 30 wherein

said process to control (488) is an internal combustion engine,

at least one of said sets of sensors first and second comprises at least one sensor belonging to the group composed of crank angle sensors, pressure sensors and water temperature sensors,

said third set of sensors comprises at least an air fuel ratio sensor.

33. The method according to one of claims from 27 to 31 wherein

said process to control (488) is a suspension vehicle,

at least one of said sets of sensors comprise at least one sensor belonging to the group composed of position sensors, pitch sensors and roll angle sensors.

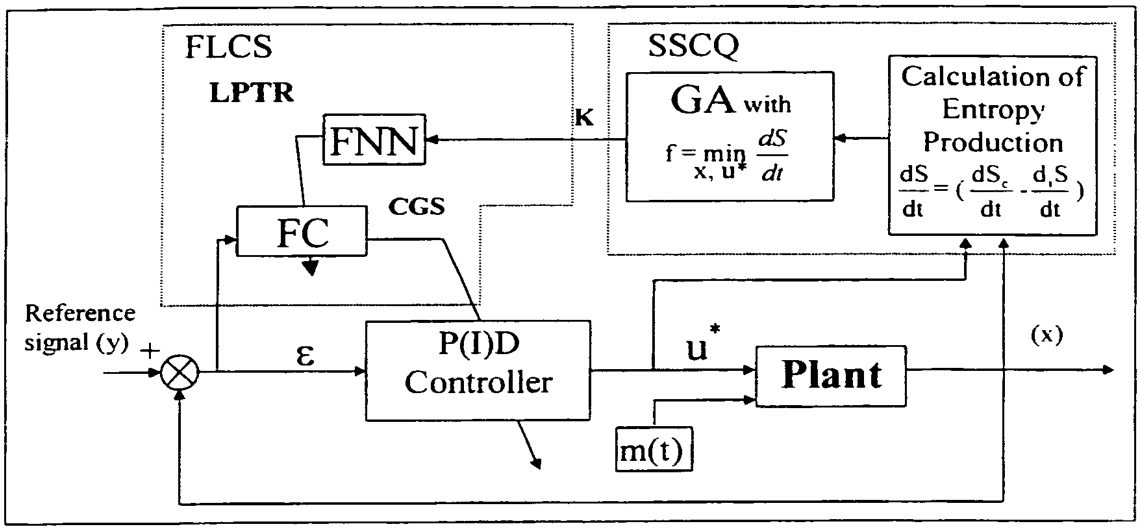

FLCS
LPTR
FNN
K
SSCQ
GA with
f = min dS/dt
x, u*
Calculation of Entropy Production
dS/dt = (dSc/dt - diS/dt)
FC
CGS
Reference signal (y)
+
ε
P(I)D Controller
u*
Plant
(x)
m(t)

Fig. 1

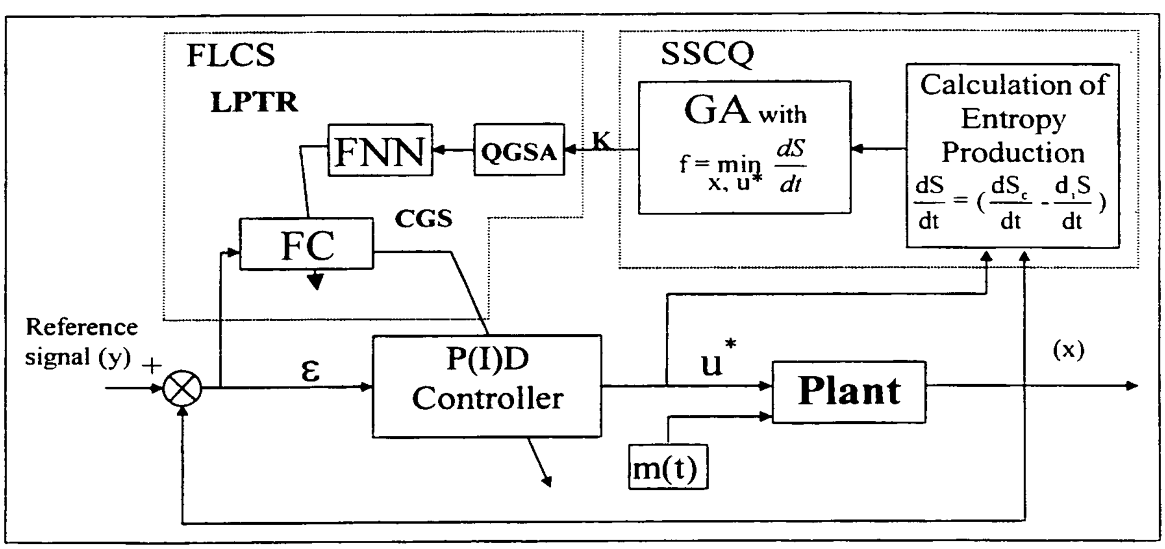

FLCS
LPTR
FNN
QGSA
K
SSCQ
GA with
f = min dS/dt
x, u*
Calculation of Entropy Production
dS/dt = (dSc/dt - diS/dt)
FC
CGS
Reference signal (y)
+
ε
P(I)D Controller
u*
Plant
(x)
m(t)

Fig. 2

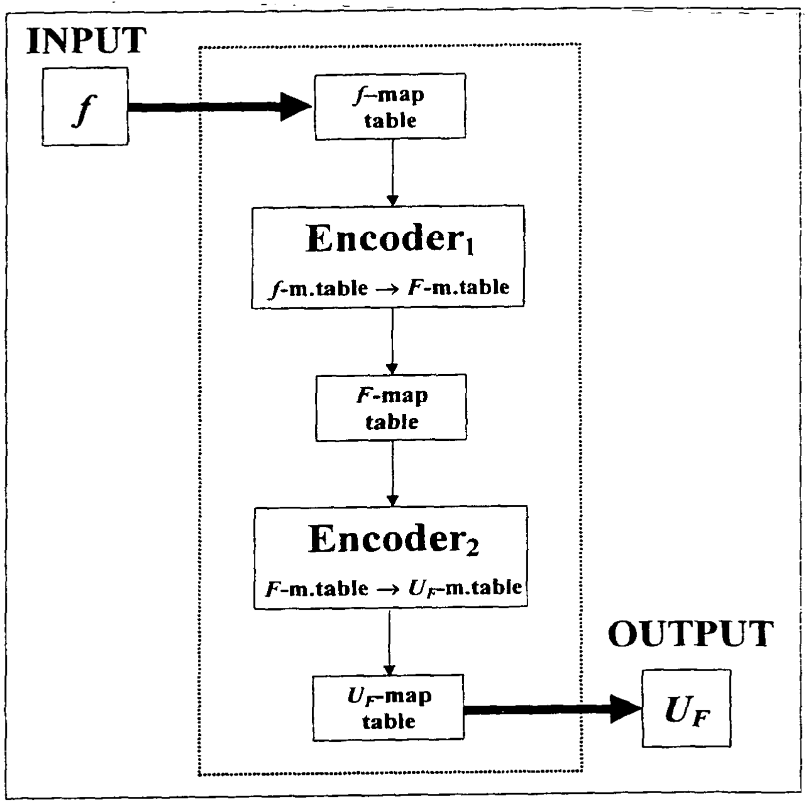

INPUT
$f$
$f$-map table
Encoder$_1$
$f$-m.table → $F$-m.table
$F$-map table
Encoder$_2$
$F$-m.table → $U_F$-m.table
$U_F$-map table
OUTPUT
$U_F$

**Fig. 3**

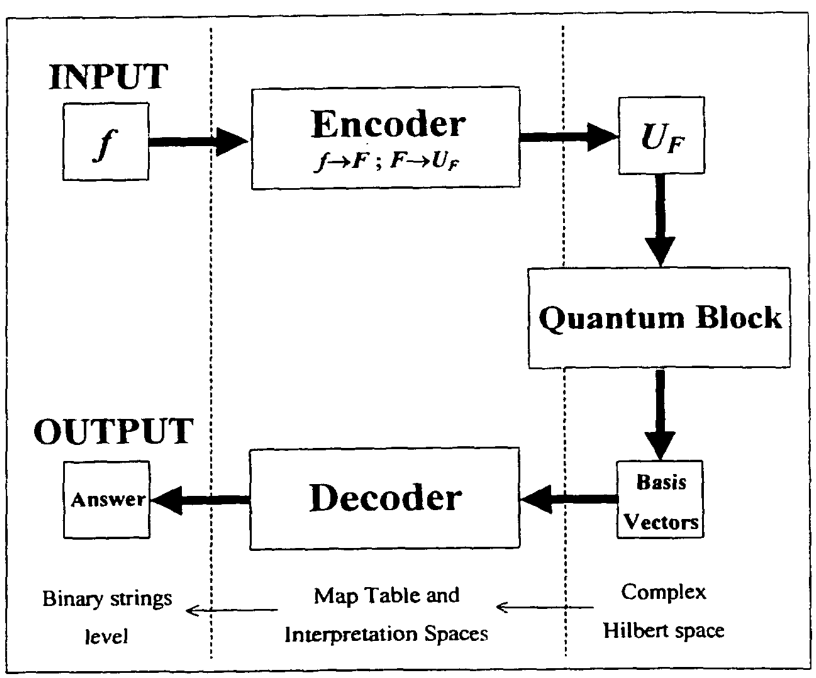

INPUT
$f$
Encoder
$f \to F$ ; $F \to U_F$
$U_F$
Quantum Block
OUTPUT
Answer
Decoder
Basis Vectors
Binary strings level
Map Table and Interpretation Spaces
Complex Hilbert space

**Fig. 4**

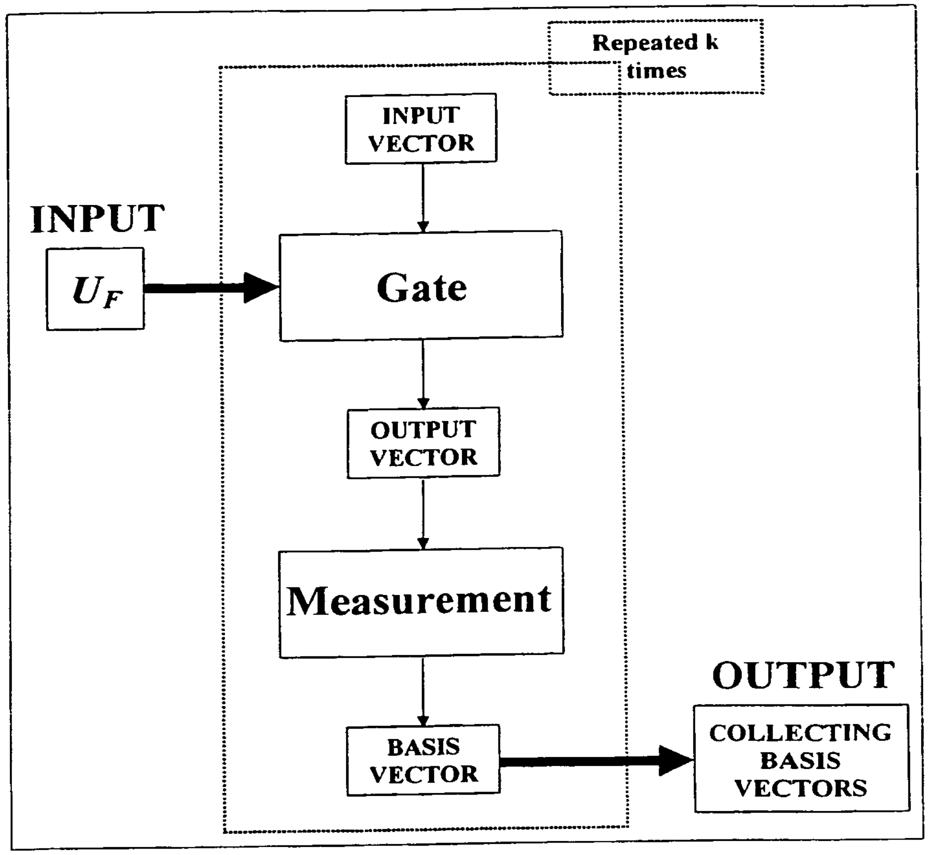

Repeated k times
INPUT VECTOR
INPUT
$U_F$
Gate
OUTPUT VECTOR
Measurement
OUTPUT
BASIS VECTOR
COLLECTING BASIS VECTORS

Fig. 5

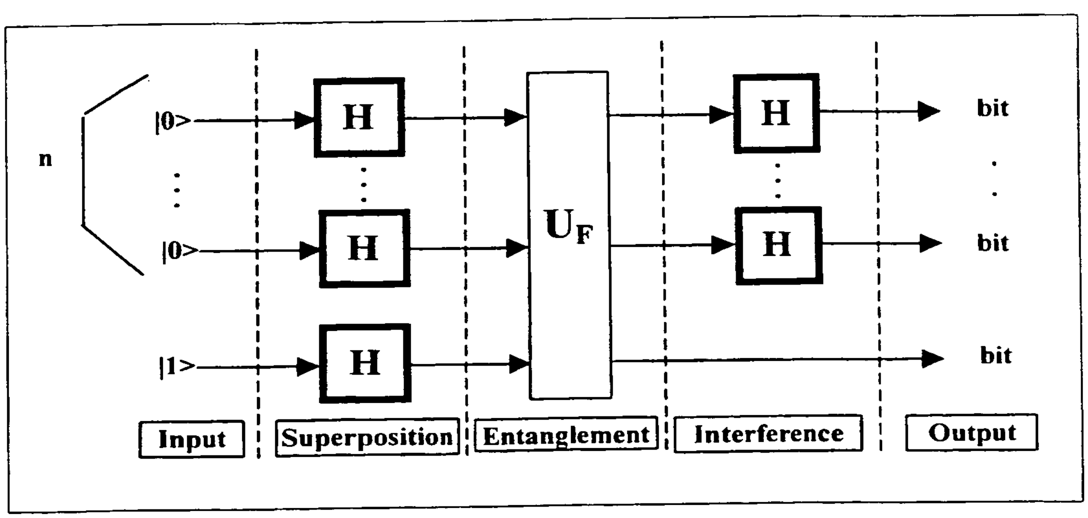

n
|0>
|0>
|1>
H
H
H
$U_F$
H
H
bit
bit
bit
Input
Superposition
Entanglement
Interference
Output

Fig. 6

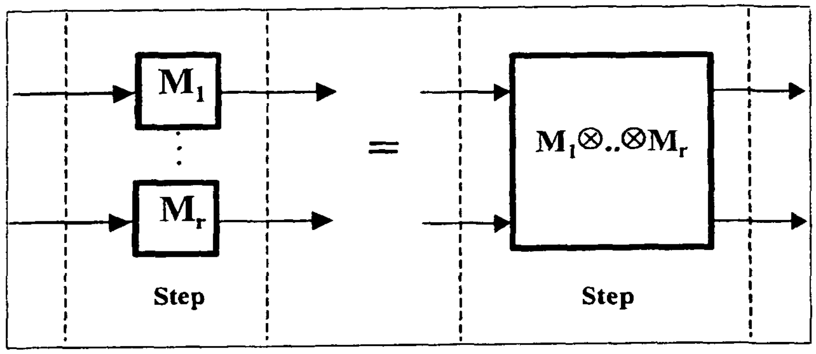



Fig. 7.a

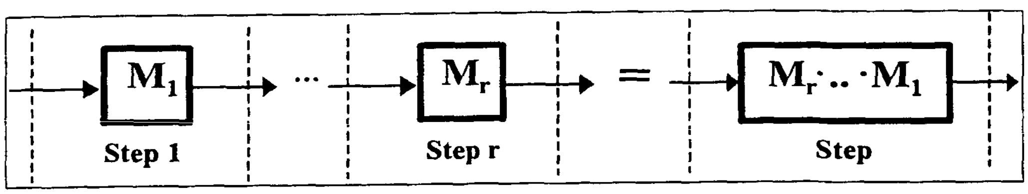



Fig. 7.b

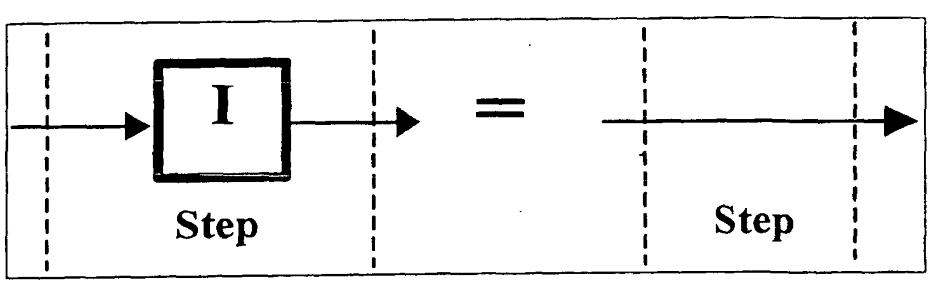



Fig. 7.c

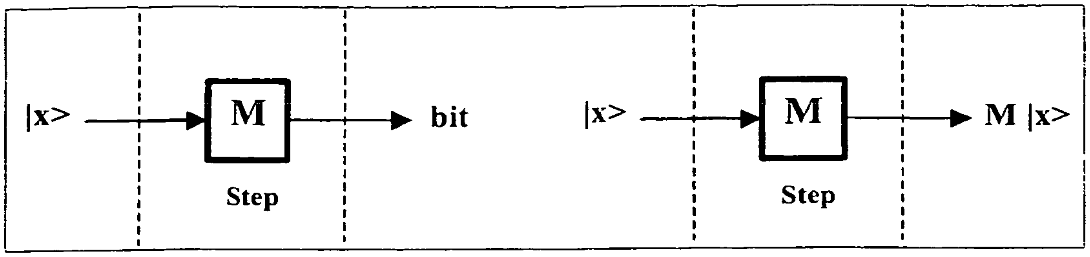



Fig. 7.d

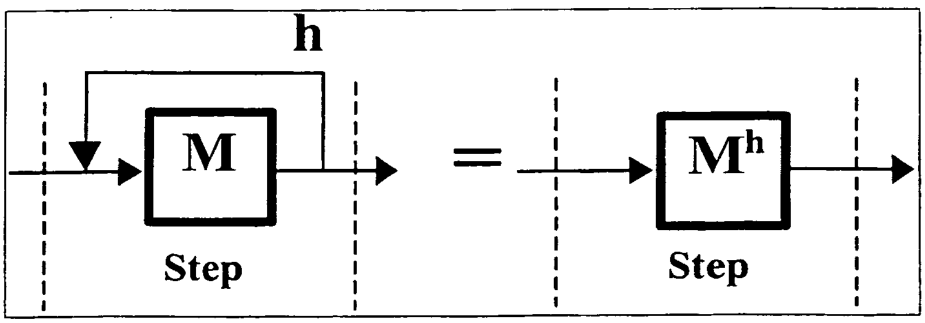



Fig. 7.e

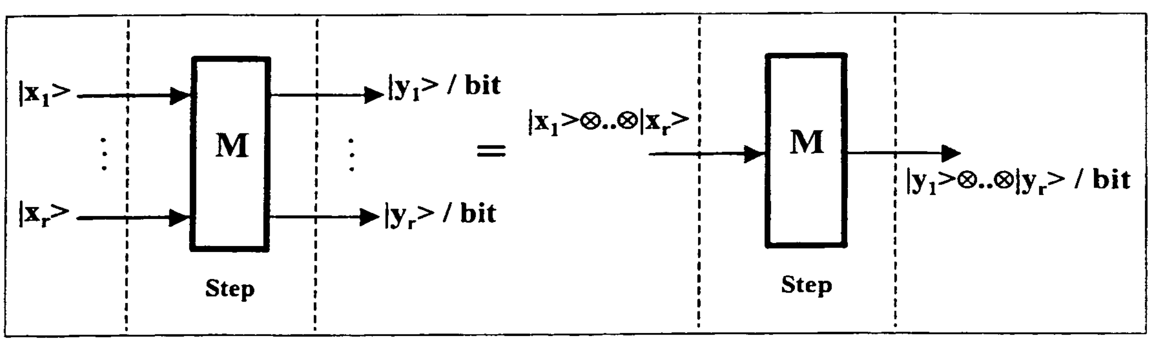



Fig. 7.f

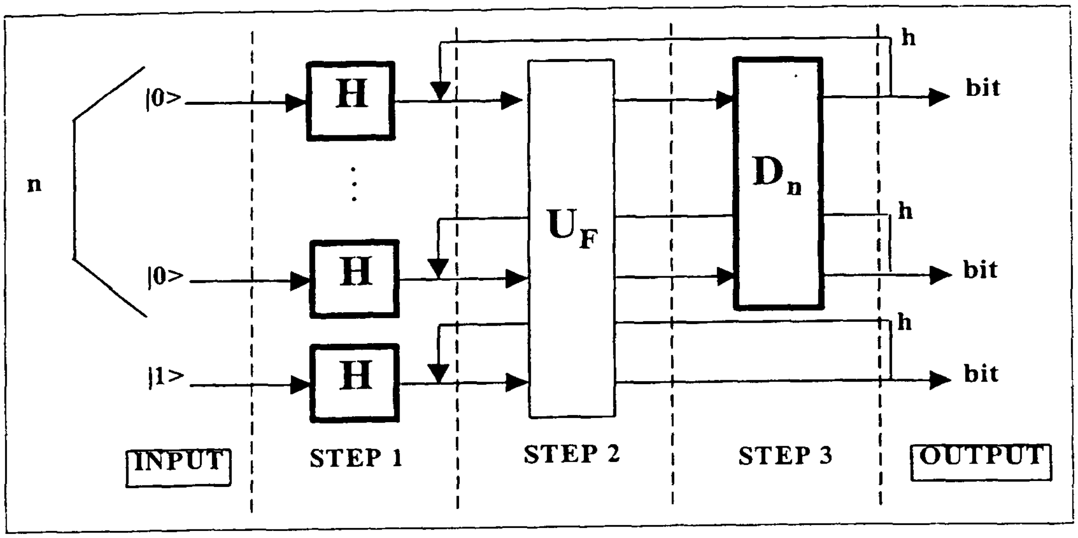

n
|0>
|0>
|1>
H
H
H
$U_F$
$D_n$
h
h
h
bit
bit
bit
INPUT
STEP 1
STEP 2
STEP 3
OUTPUT

**Fig. 8**

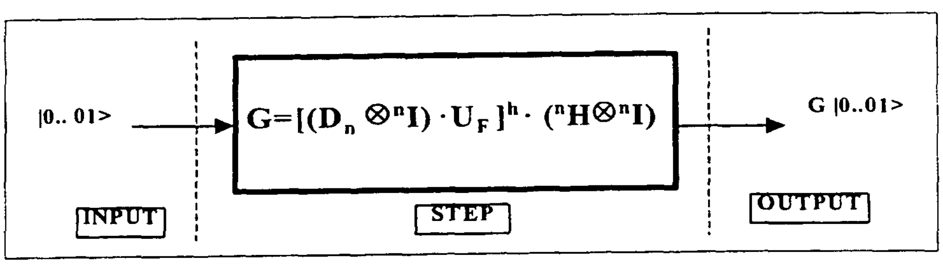

|0.. 01>
$G=[(D_n \otimes {}^nI) \cdot U_F]^h \cdot ({}^nH \otimes {}^nI)$
G |0..01>
INPUT
STEP
OUTPUT

**Fig. 9**

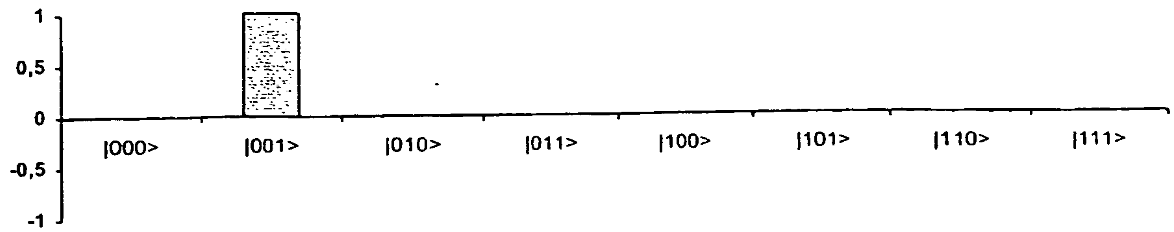



**Fig. 10.a**

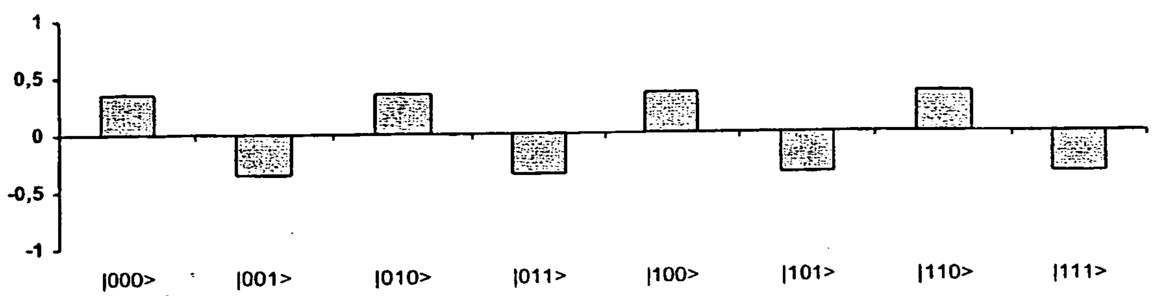



**Fig. 10.b**

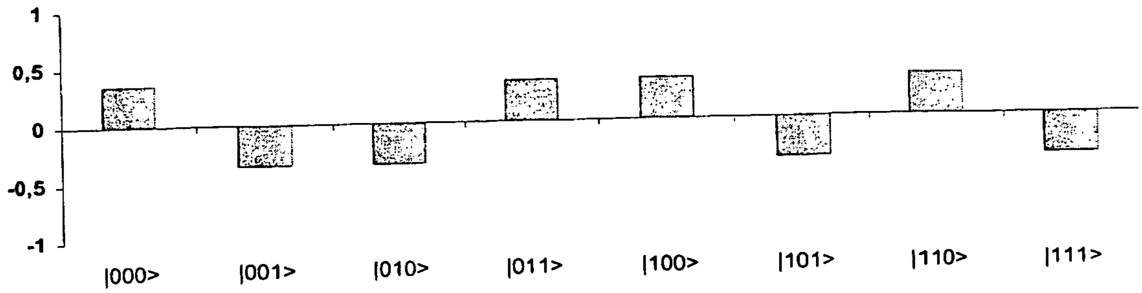



Fig. 10.c

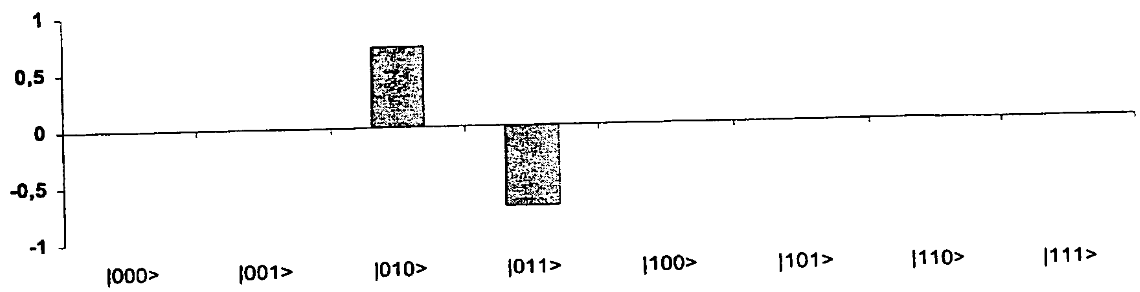



Fig. 10.d

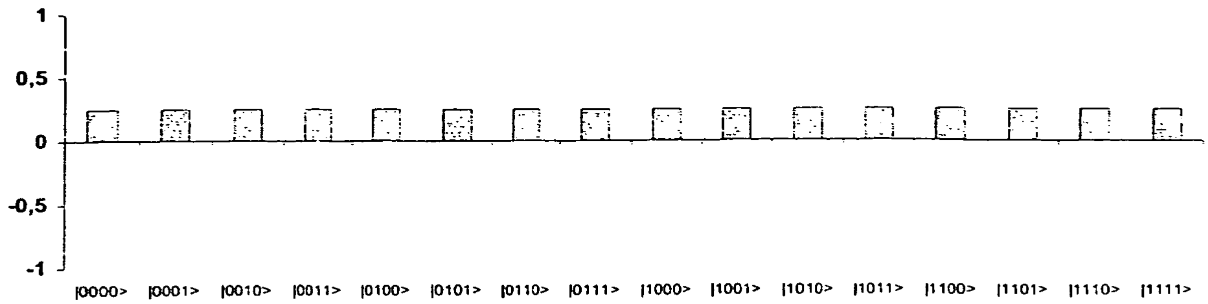



Fig. 11.a

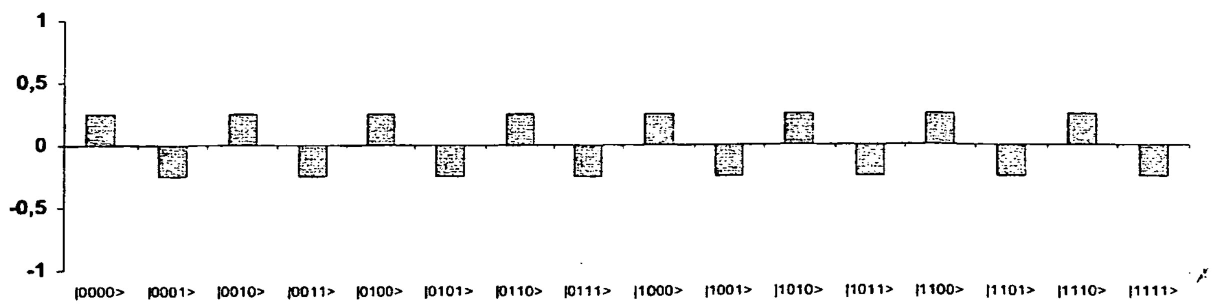



Fig. 11.b

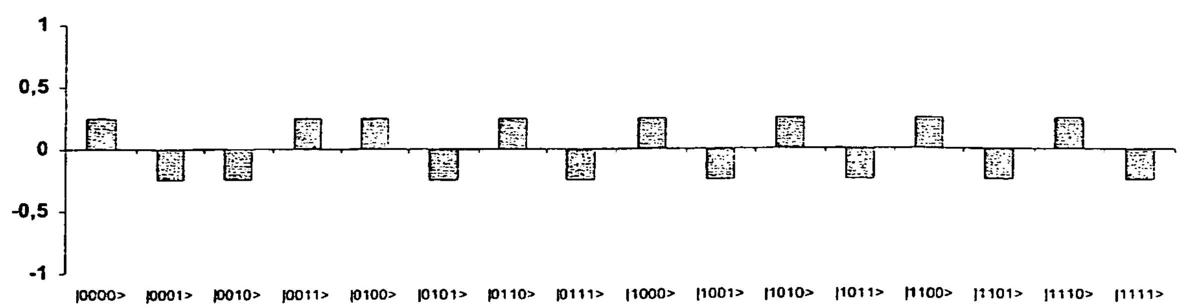



Fig. 11.c

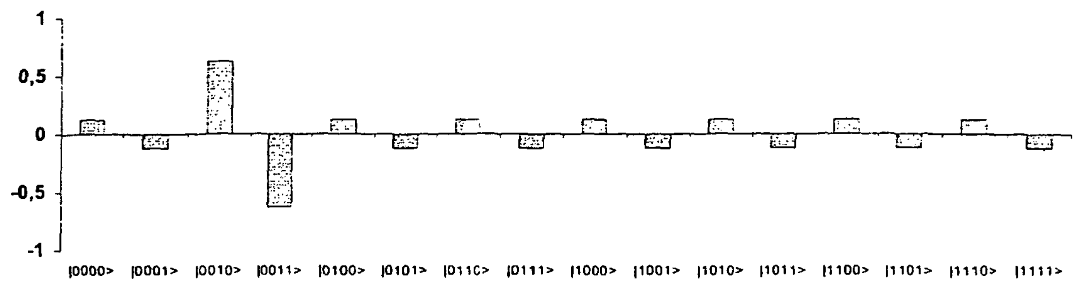



Fig. 11.d

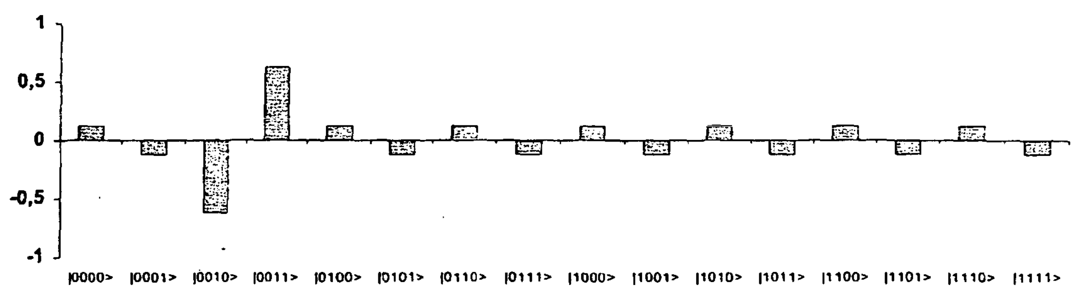



Fig. 11.e

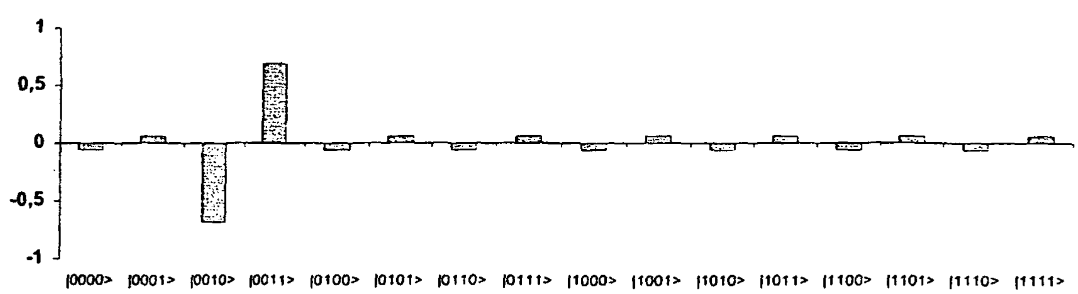



Fig. 11.f

| Step | Classical and Quantum Entropy $S_{\vert\psi\rangle}(\{j\}) = -\lambda_1 \log \lambda_1 - \lambda_2 \log \lambda_2$ | States and Gate Operations |
|---|---|---|
| Input | $\lambda_{1/2} = \frac{\alpha^2}{2}\left(-1+2^n+\frac{\beta^2}{\alpha^2} \pm \sqrt{5-2^{n+2}+2^{2n}+\left(2^{n+2}-8\right)\frac{\beta}{\alpha}+2\frac{\beta^2}{\alpha^2}+\frac{\beta^4}{\alpha^4}}\right)$ <br> $H_{Sh}(\vert\psi\rangle) = -\left(2^n-1\right)\alpha^2 \log\alpha^2 - \beta^2 \log\beta^2$ | $\alpha\left(\vert 0...0\rangle + ... + \frac{\beta}{\alpha}\vert \underline{x}\rangle + ... + \vert 1...1\rangle\right) \otimes$ <br> $\otimes \frac{\vert 0\rangle - \vert 1\rangle}{\sqrt{2}}$ |
| Entangle-ment | $\lambda_{1/2} = \frac{\alpha^2}{2}\left(-1+2^n+\frac{\beta^2}{\alpha^2} \pm \sqrt{5-2^{n+2}+2^{2n}-\left(2^{n+2}-8\right)\frac{\beta}{\alpha}+2\frac{\beta^2}{\alpha^2}+\frac{\beta^4}{\alpha^4}}\right)$ <br> $H_{Sh}(\vert\psi\rangle) = -\left(2^n-1\right)\alpha^2 \log\alpha^2 - \beta^2 \log\beta^2$ | $\alpha\left(\vert 0...0\rangle + ... - \frac{\beta}{\alpha}\vert \underline{x}\rangle + ... + \vert 1...1\rangle\right) \otimes \frac{\vert 0\rangle - \vert 1\rangle}{\sqrt{2}}$ |
| Interference | $\lambda_{1/2} = \frac{(\alpha-m)^2}{2}\left(-1+2^n+\frac{(\beta+m)^2}{(\alpha-m)^2} \pm \sqrt{5-2^{n+2}+2^{2n}-\left(2^{n+2}-8\right)\frac{\beta+m}{\alpha-m}+2\left(\frac{\beta+m}{\alpha-m}\right)^2+\frac{(\beta+m)^4}{(\alpha-m)^4}}\right)$ <br> $H_{Sh}(\vert\psi\rangle) = -\left(2^n-1\right)(\alpha-m)^2 \log(\alpha-m)^2 - (\beta+m)^2 \log(\beta+m)^2$ | $(\alpha-m)\left(\vert 0...0\rangle + ... + \frac{\beta+m}{\alpha-m}\vert \underline{x}\rangle + ... + \vert 1...1\rangle\right)$ <br> $\otimes \frac{\vert 0\rangle - \vert 1\rangle}{\sqrt{2}}$ |

Table 1

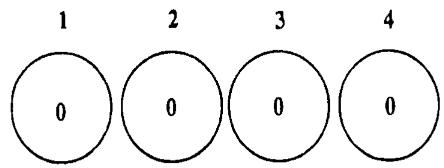
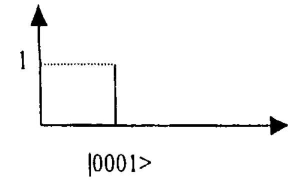
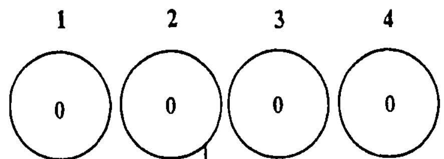
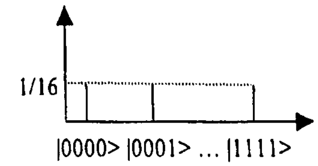
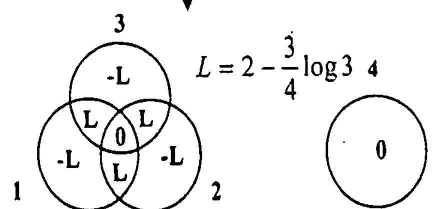
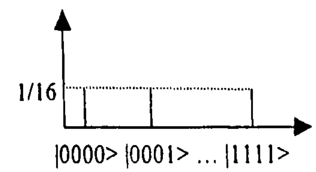
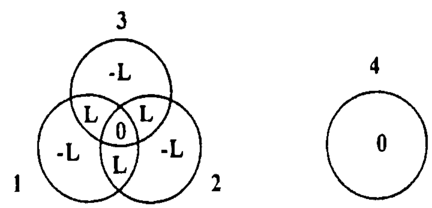
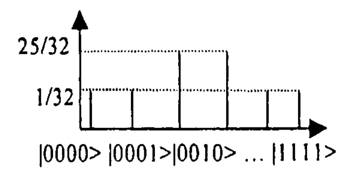
| Step | Classical Entropy $H_{Sh}(\lvert\psi\rangle)$ | Quantum Entropy $S_{\lvert\psi\rangle}(\{j\})$ | States and Gate Operations | Dynamics of Solution Probabilities |
|---|---|---|---|---|
| Input | 0 |   | $\lvert 000\rangle \otimes \lvert 1\rangle$ $\xrightarrow{{}^4H}$ |   |
| Super-position | 4 |   | $\frac{\lvert 0\rangle+\lvert 1\rangle}{\sqrt{2}} \otimes \frac{\lvert 0\rangle+\lvert 1\rangle}{\sqrt{2}} \otimes \frac{\lvert 0\rangle+\lvert 1\rangle}{\sqrt{2}} \otimes \frac{\lvert 0\rangle-\lvert 1\rangle}{\sqrt{2}}$ $\xrightarrow{U_F}$ |   |
| Entan-glement | 4 |   $L = 2 - \frac{3}{4}\log 3$ | $\frac{1}{2\sqrt{2}}\begin{pmatrix}\lvert 000\rangle - \lvert 001\rangle + \lvert 010\rangle + \\ \lvert 011\rangle + \lvert 100\rangle + \lvert 101\rangle + \\ + \lvert 110\rangle + \lvert 111\rangle\end{pmatrix} \otimes \frac{\lvert 0\rangle-\lvert 1\rangle}{\sqrt{2}}$ $\xrightarrow{D_3 \otimes I}$ |   |
| Inter-ference | $5 - \frac{25}{16}\log 5$ | $L = -\frac{1}{8}(4-\sqrt{13})\log\left[\frac{1}{8}(4-\sqrt{13})\right] - \frac{1}{8}(4+\sqrt{13})\log\left[\frac{1}{8}(4+\sqrt{13})\right]$   | $\frac{1}{4\sqrt{2}}\begin{pmatrix}\lvert 000\rangle + 5\lvert 001\rangle + \lvert 010\rangle + \\ \lvert 011\rangle + \lvert 100\rangle + \lvert 101\rangle + \\ + \lvert 110\rangle + \lvert 111\rangle\end{pmatrix} \otimes \frac{\lvert 0\rangle-\lvert 1\rangle}{\sqrt{2}}$ |   |

Table 2

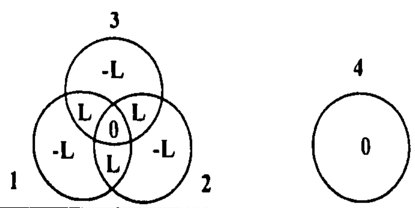
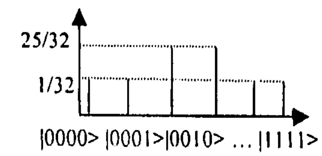
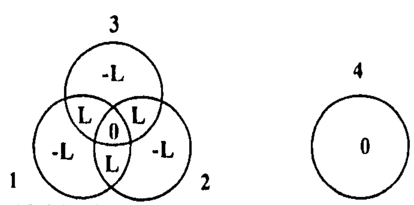
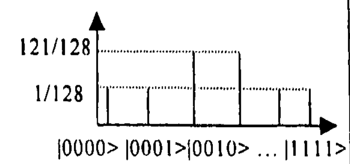
| Step | Classical Entropy $H_{Sh}(\|\psi\rangle)$ | Quantum Entropy $S_{\|\psi\rangle}(\{j\})$ | States and Gate Operations | Dynamics of Solution Probabilities |
|---|---|---|---|---|
| Entan-glement | $5-\frac{25}{16}\log 5$ | $L=-\frac{1}{16}(8-\sqrt{37})\log\left[\frac{1}{16}(8-\sqrt{37})\right]-\frac{1}{16}(8+\sqrt{37})\log\left[\frac{1}{16}(8+\sqrt{37})\right]$   | $\boxed{U_F}$ $\frac{1}{4\sqrt{2}}\begin{pmatrix}\|000\rangle-5\|001\rangle+\|010\rangle+\\ \|011\rangle+\|100\rangle+\|101\rangle+\\ +\|110\rangle+\|111\rangle\end{pmatrix}\otimes\frac{\|0\rangle-\|1\rangle}{\sqrt{2}}$ |   |
| Inter-ference | $6-\frac{121}{64}\log 11$ | $L=-\frac{1}{256}(128-\sqrt{14656})\log\left[\frac{1}{256}(128-\sqrt{14656})\right]-$ $-\frac{1}{256}(128+\sqrt{14656})\log\left[\frac{1}{256}(128+\sqrt{14656})\right]$   | $\boxed{D_3\otimes I}$ $-\frac{1}{8\sqrt{2}}\begin{pmatrix}\|000\rangle-11\|001\rangle+\|010\rangle+\\ \|011\rangle+\|100\rangle+\|101\rangle+\\ +\|110\rangle+\|111\rangle\end{pmatrix}\otimes\frac{\|0\rangle-\|1\rangle}{\sqrt{2}}$ |   |

Table 3

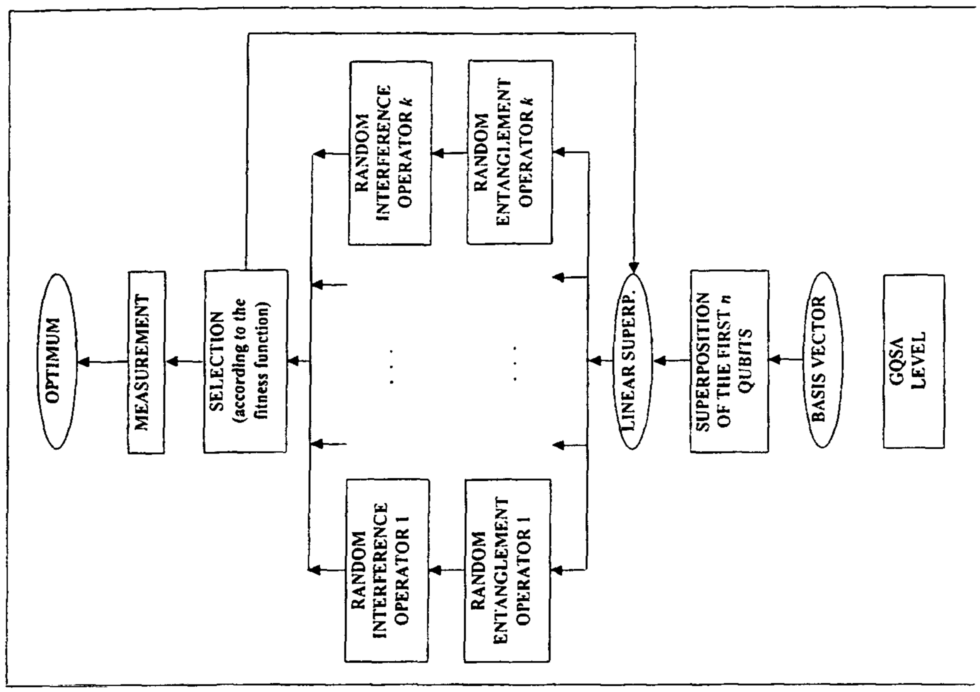

OPTIMUM
MEASUREMENT
SELECTION (according to the fitness function)
RANDOM INTERFERENCE OPERATOR k
RANDOM ENTANGLEMENT OPERATOR k
RANDOM INTERFERENCE OPERATOR 1
RANDOM ENTANGLEMENT OPERATOR 1
LINEAR SUPERP.
SUPERPOSITION OF THE FIRST n QUBITS
BASIS VECTOR
GQSA LEVEL

Fig. 13

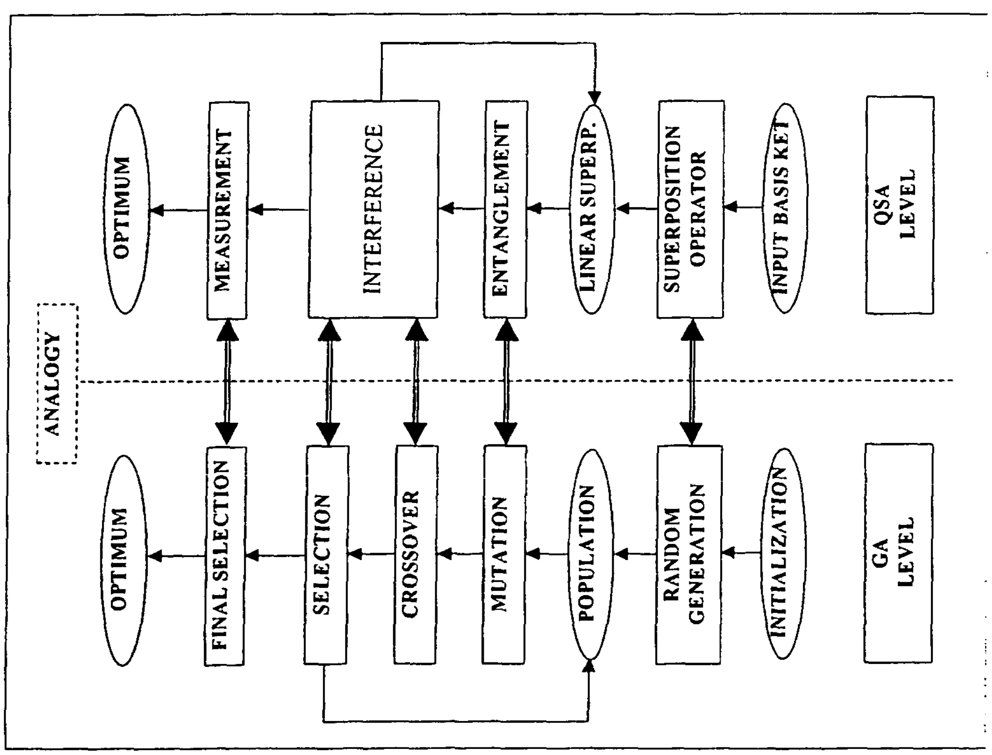
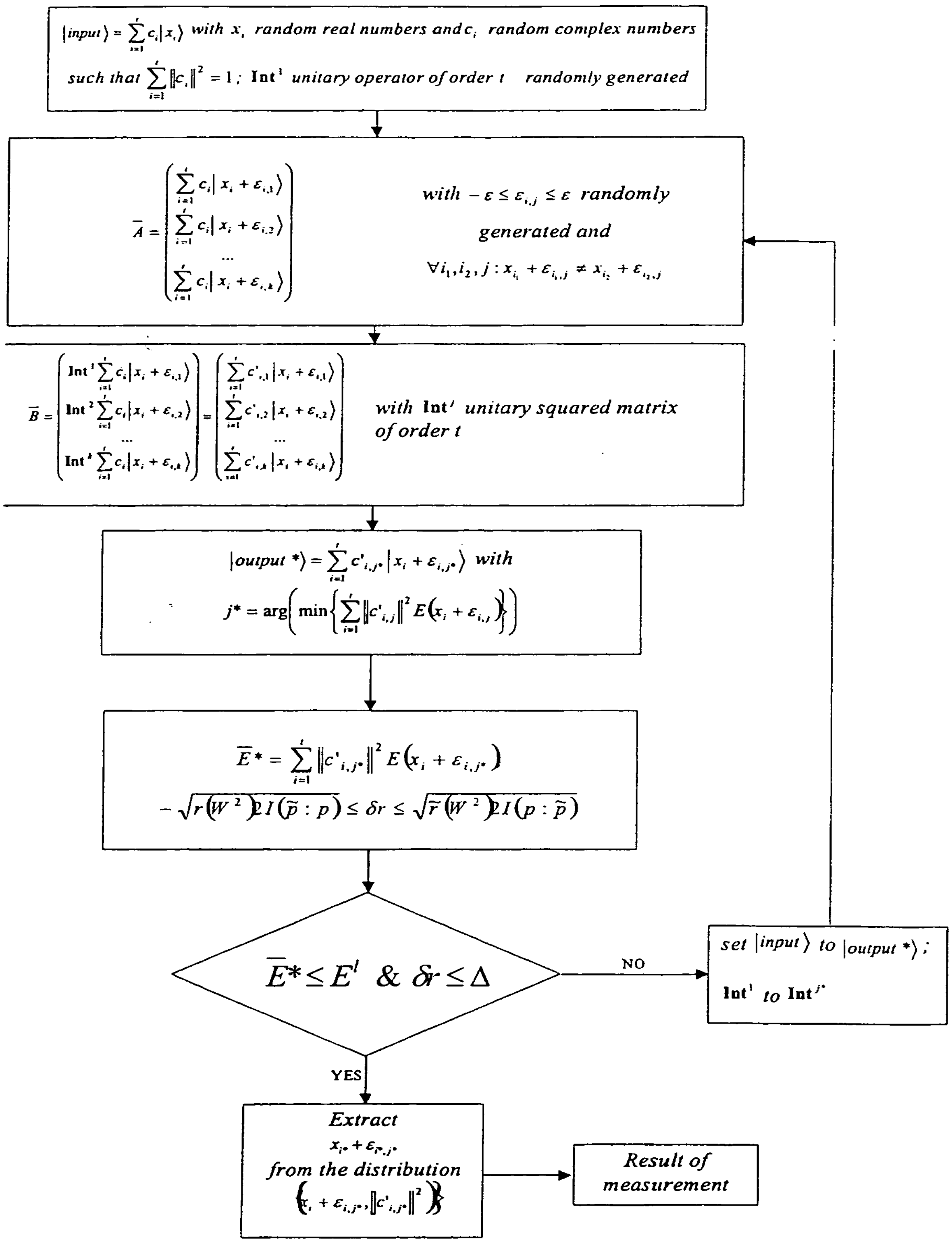

ANALOGY
OPTIMUM
MEASUREMENT
INTERFERENCE
ENTANGLEMENT
LINEAR SUPERP.
SUPERPOSITION OPERATOR
INPUT BASIS KET
QSA LEVEL
OPTIMUM
FINAL SELECTION
SELECTION
CROSSOVER
MUTATION
POPULATION
RANDOM GENERATION
INITIALIZATION
GA LEVEL

Fig. 12

**Fig. 14**


$|input\rangle = \sum_{i=1}^{t} c_i|x_i\rangle$ with $x_i$ random real numbers and $c_i$ random complex numbers such that $\sum_{i=1}^{t}\|c_i\|^2 = 1$; $\mathbf{Int}^1$ unitary operator of order $t$ randomly generated
$\overline{A} = \begin{pmatrix} \sum_{i=1}^{t} c_i|x_i + \varepsilon_{i,1}\rangle \\ \sum_{i=1}^{t} c_i|x_i + \varepsilon_{i,2}\rangle \\ \dots \\ \sum_{i=1}^{t} c_i|x_i + \varepsilon_{i,k}\rangle \end{pmatrix}$
with $-\varepsilon \le \varepsilon_{i,j} \le \varepsilon$ randomly generated and $\forall i_1, i_2, j : x_{i_1} + \varepsilon_{i_1,j} \ne x_{i_2} + \varepsilon_{i_2,j}$
$\overline{B} = \begin{pmatrix} \mathbf{Int}^1 \sum_{i=1}^{t} c_i|x_i + \varepsilon_{i,1}\rangle \\ \mathbf{Int}^2 \sum_{i=1}^{t} c_i|x_i + \varepsilon_{i,2}\rangle \\ \dots \\ \mathbf{Int}^k \sum_{i=1}^{t} c_i|x_i + \varepsilon_{i,k}\rangle \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{t} c'_{i,1}|x_i + \varepsilon_{i,1}\rangle \\ \sum_{i=1}^{t} c'_{i,2}|x_i + \varepsilon_{i,2}\rangle \\ \dots \\ \sum_{i=1}^{t} c'_{i,k}|x_i + \varepsilon_{i,k}\rangle \end{pmatrix}$
with $\mathbf{Int}^j$ unitary squared matrix of order $t$
$|output^*\rangle = \sum_{i=1}^{t} c'_{i,j^*}|x_i + \varepsilon_{i,j^*}\rangle$ with
$j^* = \arg\left(\min\left\{\sum_{i=1}^{t}\|c'_{i,j}\|^2 E(x_i + \varepsilon_{i,j})\right\}\right)$
$\overline{E}^* = \sum_{i=1}^{t}\|c'_{i,j^*}\|^2 E(x_i + \varepsilon_{i,j^*})$
$-\sqrt{r(W^2)2I(\tilde{p}:p)} \le \delta r \le \sqrt{\tilde{r}(W^2)2I(p:\tilde{p})}$
$\overline{E}^* \le E^l \;\&\; \delta r \le \Delta$
NO
set $|input\rangle$ to $|output^*\rangle$;
$\mathbf{Int}^1$ to $\mathbf{Int}^{j^*}$
YES
Extract
$x_{i^*} + \varepsilon_{i^*,j^*}$
from the distribution
$\{x_i + \varepsilon_{i,j^*}, \|c'_{i,j^*}\|^2\}$
Result of measurement

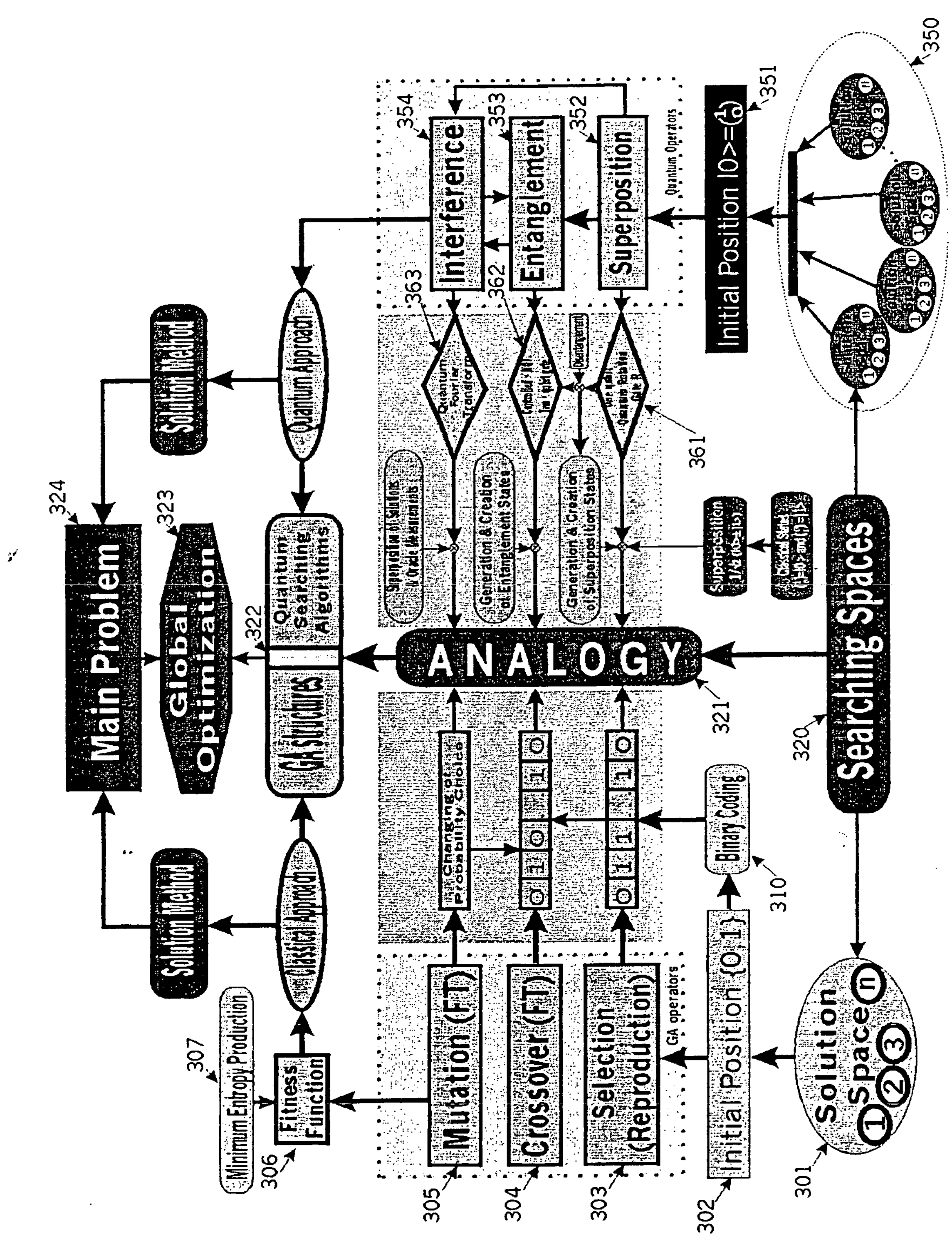

Main Problem
Global Optimization
GA Structures
Quantum Searching Algorithms
Solution Method
Solution Method
Classical Approach
Quantum Approach
Minimum Entropy Production
Fitness Function
ANALOGY
Mutation (FT)
Crossover (FT)
Selection (Reproduction)
GA operators
Changing of Probability Choice
Binary Coding
Initial Position {0,1}
Solution Space
Searching Spaces
Interference
Entanglement
Superposition
Quantum Operators
Quantum Fourier Transform
Generation & Creation of Entanglement States
Generation & Creation of Superposition States
Superposition
Initial Position
324
323
322
321
320
307
306
305
304
303
302
301
310
350
351
352
353
354
361
362
363

Fig. 15

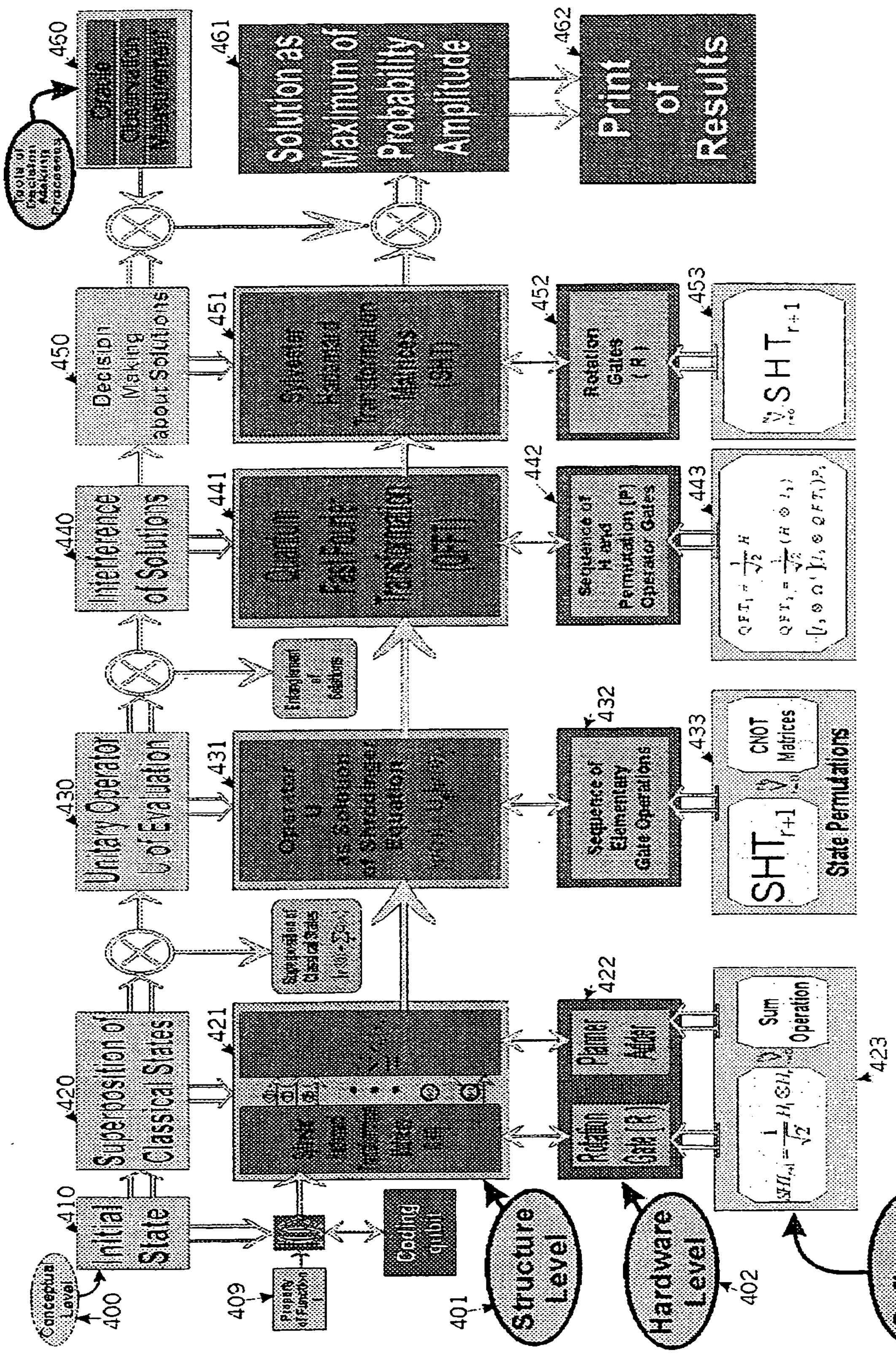

Conceptual Level
400
Initial State
410
Superposition of Classical States
420
Unitary Operator U of Evaluation
430
Interference of Solutions
440
Decision Making about Solutions
450
Oracle
Observation
Measurement
450
Solution as Maximum of Probability Amplitude
461
Print of Results
462
Property of Function f
409
Coding qubit
421
Operator U as Solution of Shrödinger Equation
431
Quantum Fast Fourier Transformation (QFT)
441
Symmetric Hadamard Transformation Matrices (SHT)
451
Entanglement of Solutions
Superposition of Classical States
Structure Level
401
Hardware Level
402
Software Level
403
Planner Adder
Rotation Gate (R)
422
Sequence of Elementary Gate Operations
432
Sequence of Hard Permutation (P) Operator Gates
442
Rotation Gates (R)
452
423
Sum Operation
433
CNOT Matrices
State Permutations
443
453

Fig. 16

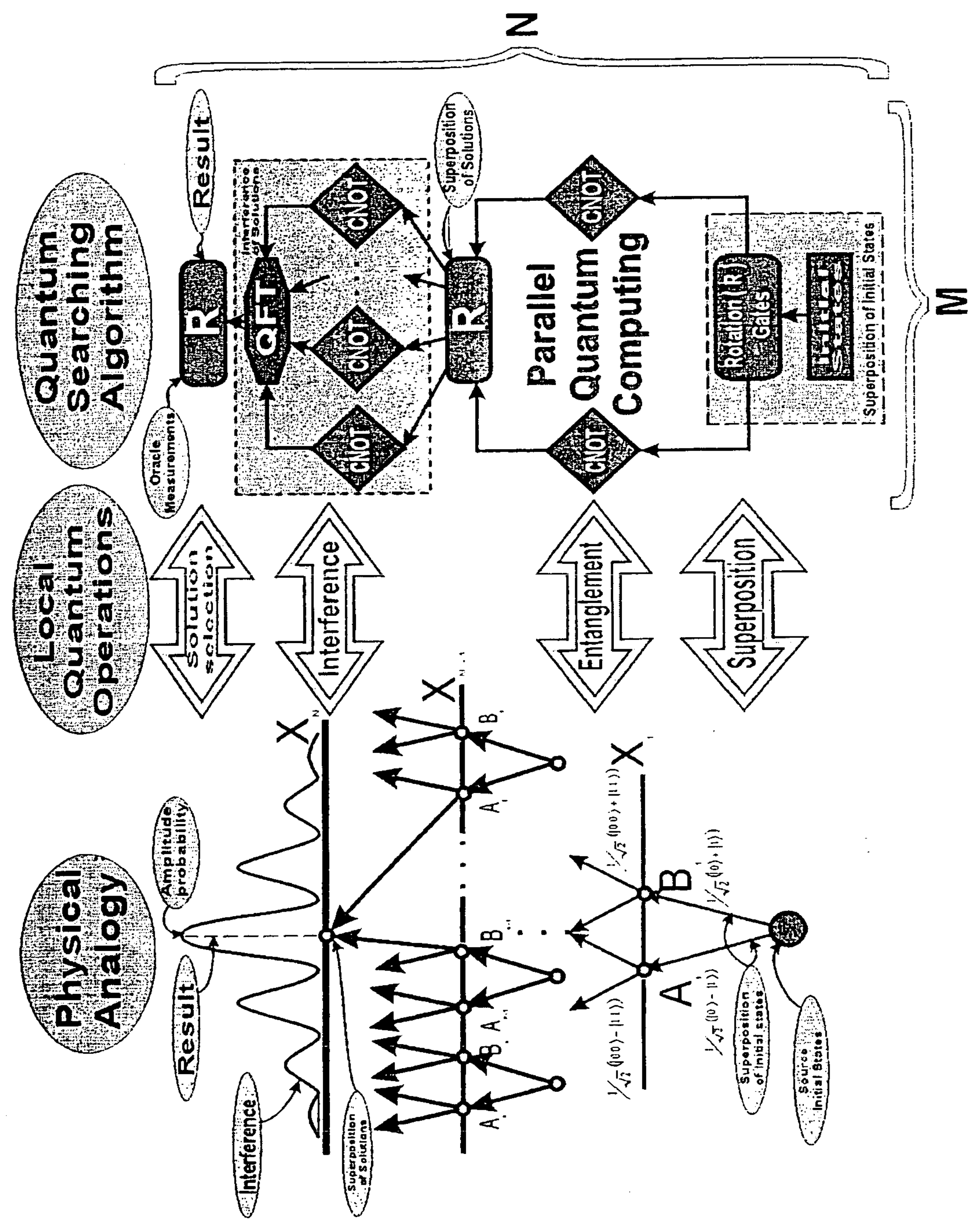

Physical Analogy
Local Quantum Operations
Quantum Searching Algorithm
Result
Amplitude probability
Interference
Superposition of Solutions
Superposition of initial states
Source Initial States
Solution selection
Interference
Entanglement
Superposition
Oracle Measurements
R
QFT
CNOT
Interference of Solutions
Superposition of Solutions
Parallel Quantum Computing
Rotation (R) Gates
Superposition of Initial States
N
M
A
B

Fig. 17

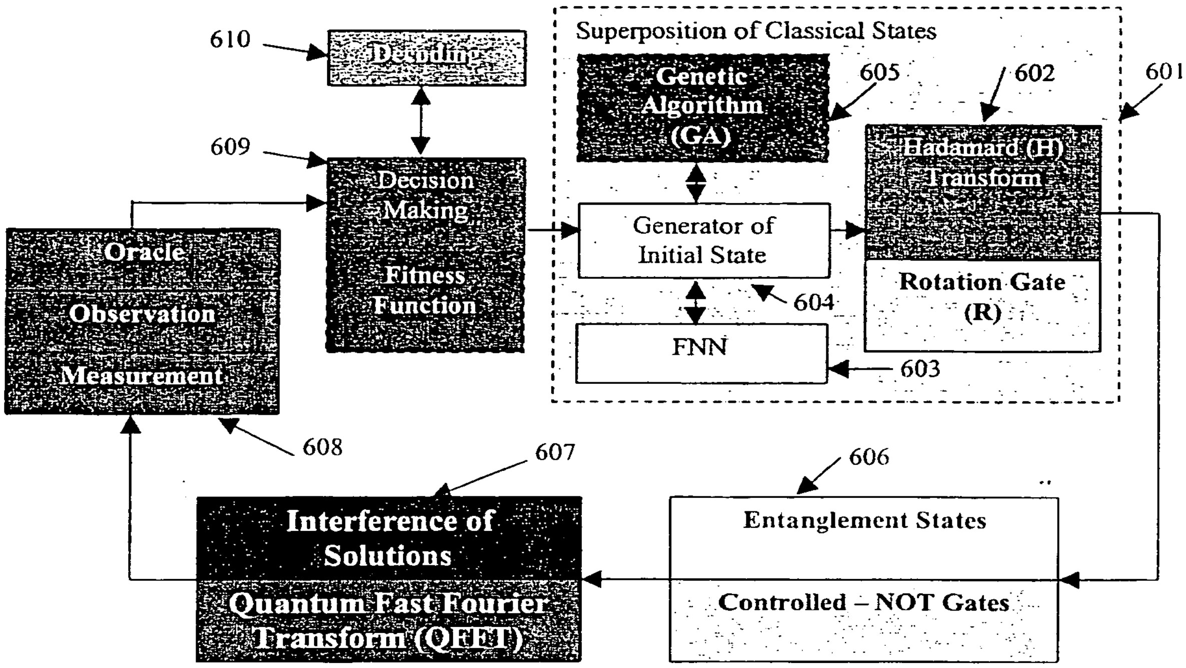
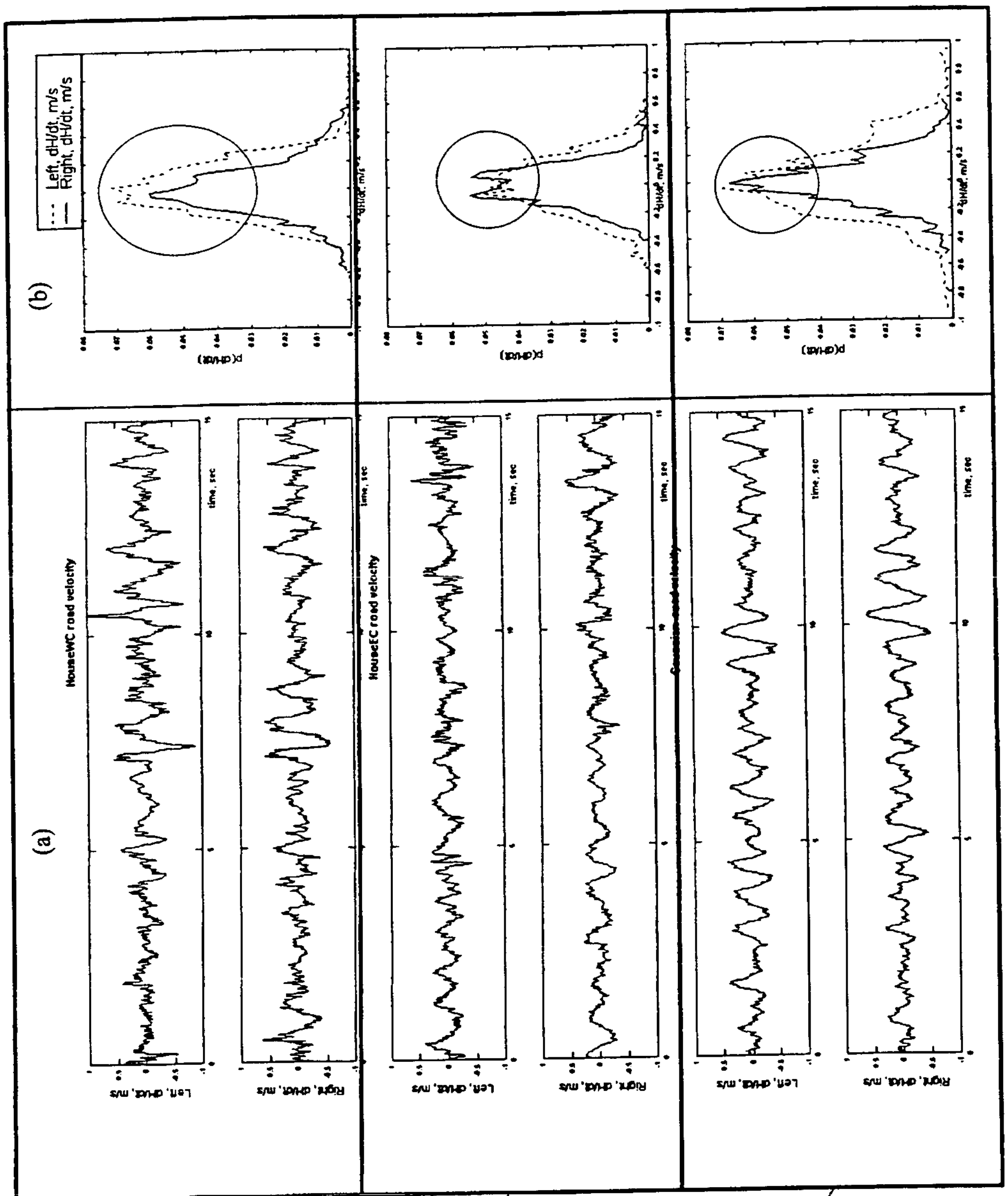

Superposition of Classical States
Decoding
Decision Making
Fitness Function
Genetic Algorithm (GA)
Generator of Initial State
FNN
Hadamard (H) Transform
Rotation Gate (R)
Oracle
Observation
Measurement
Interference of Solutions
Quantum Fast Fourier Transform (QFFT)
Entanglement States
Controlled – NOT Gates
601
602
603
604
605
606
607
608
609
610

Fig. 18




1901

1902

1903

Fig. 19

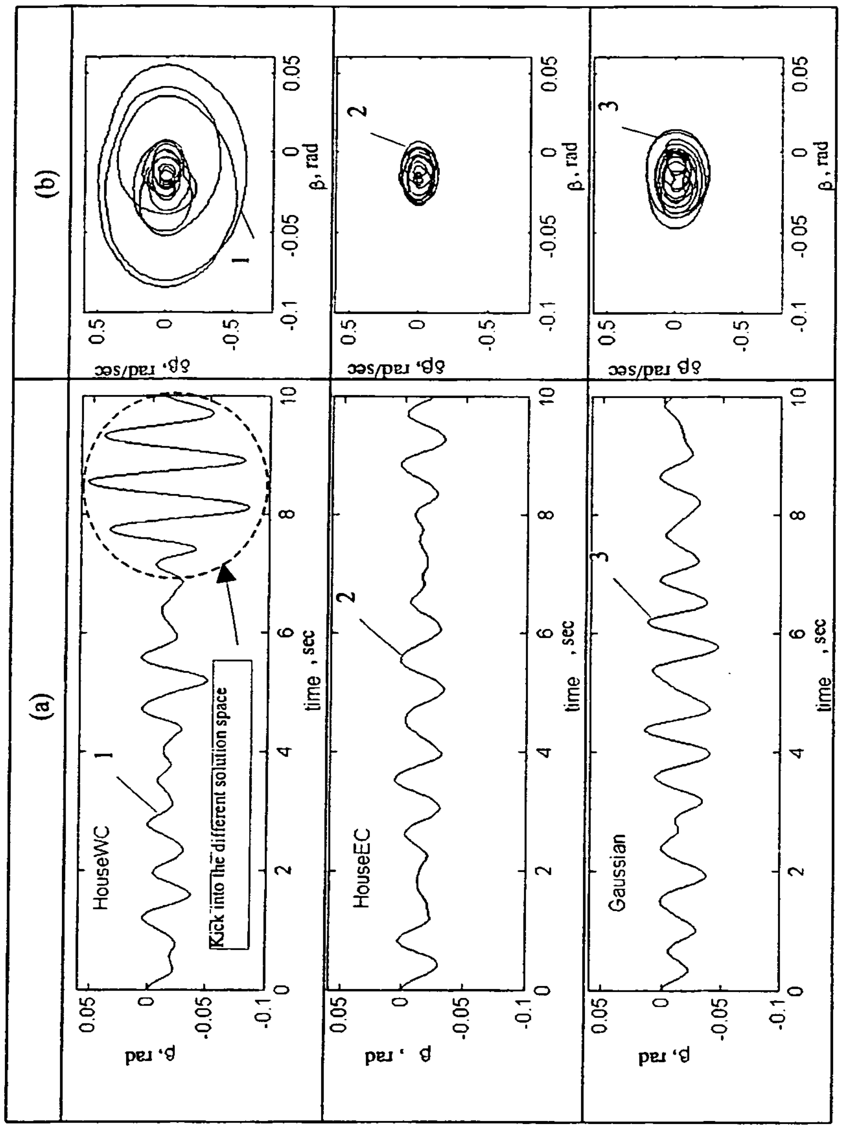

(a)
(b)
HouseWC
Kick into the different solution space
HouseEC
Gaussian
time , sec
β, rad
δβ, rad/sec
1
2
3

Fig. 20

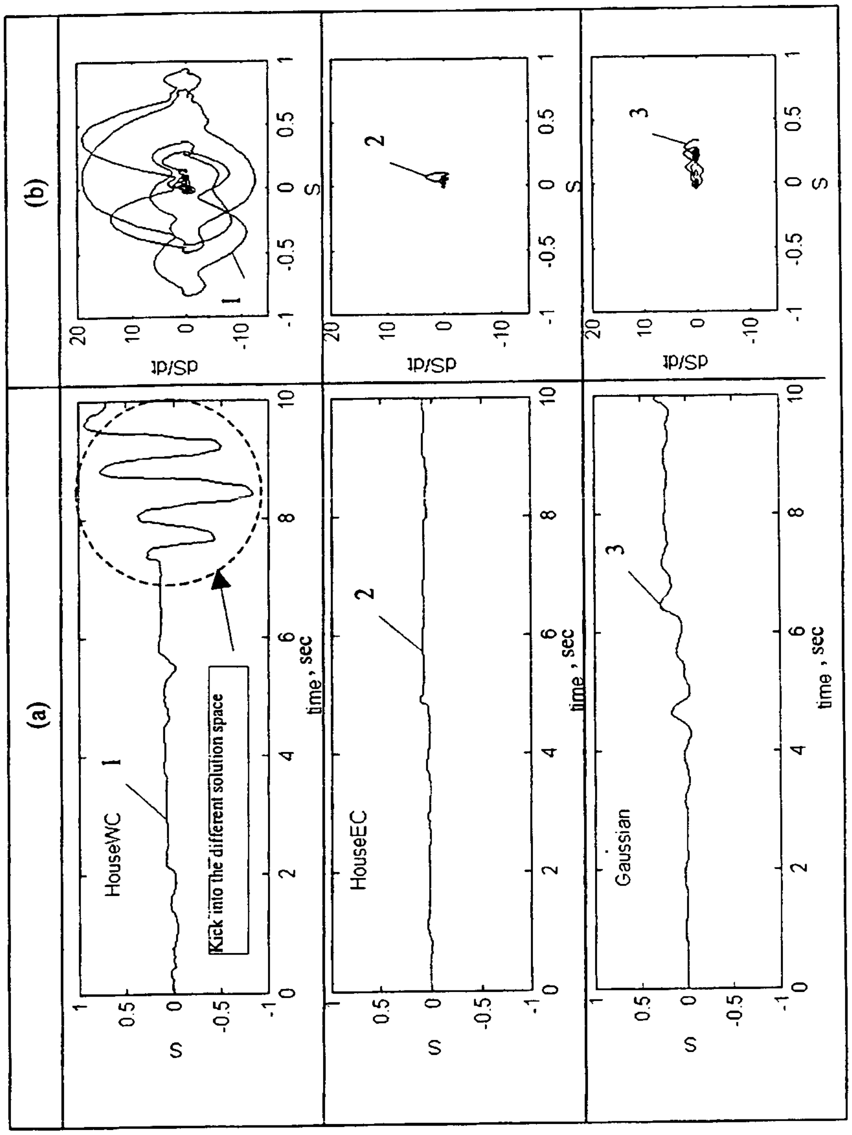

(a)
(b)
HouseWC
HouseEC
Gaussian
Kick into the different solution space
time , sec
S
dS/dt
1
2
3

Fig. 21

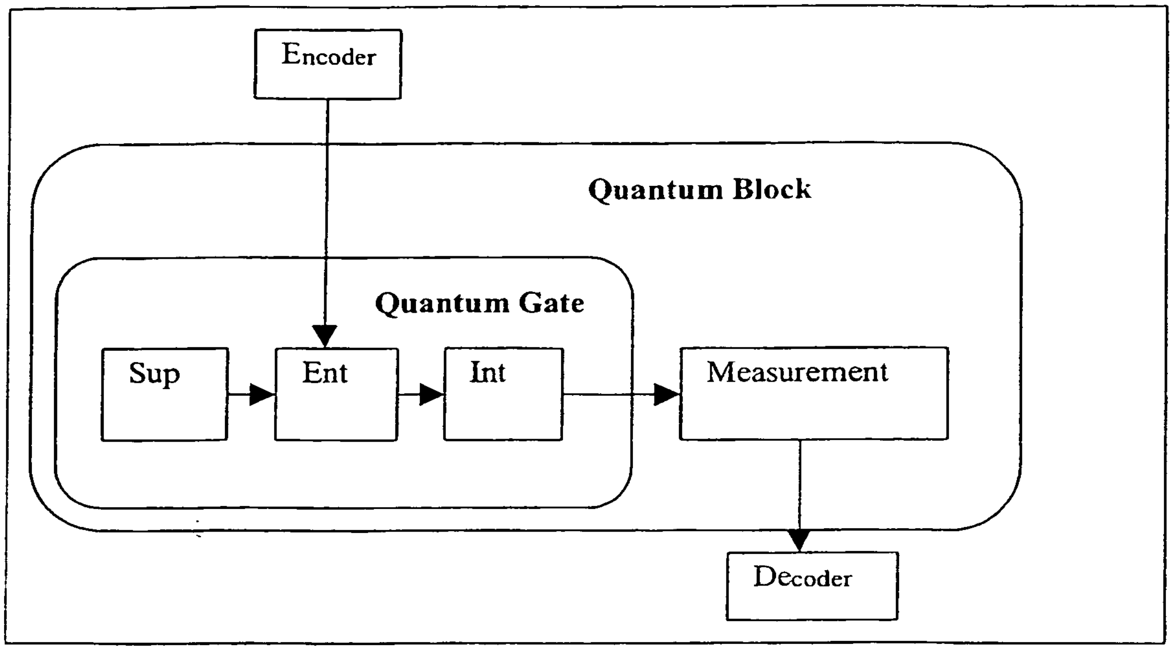

Encoder
Quantum Block
Quantum Gate
Sup
Ent
Int
Measurement
Decoder

**Fig. 22**

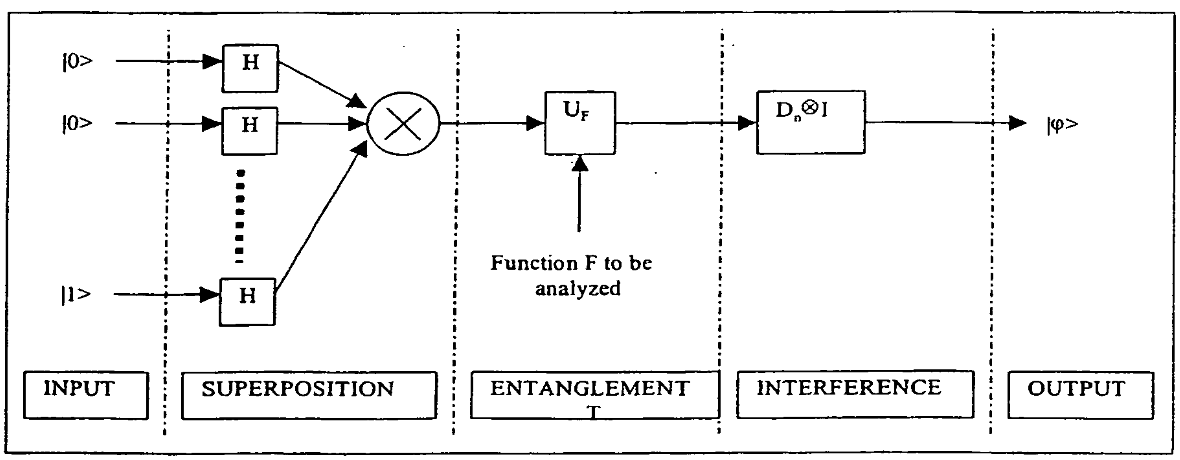

|0>
|0>
|1>
H
H
H
U$_F$
D$_n$⊗I
|φ>
Function F to be analyzed
INPUT
SUPERPOSITION
ENTANGLEMENT
INTERFERENCE
OUTPUT

**Fig. 23**

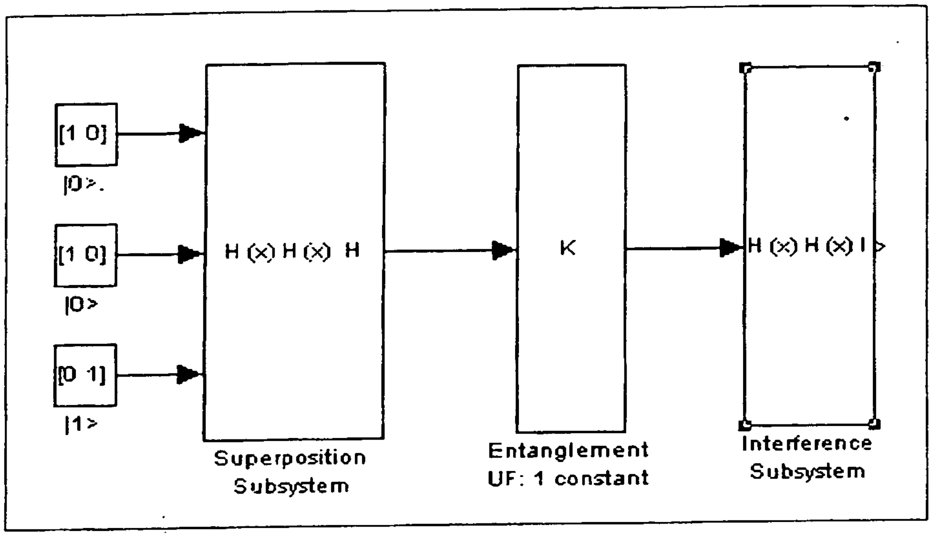



Fig. 24

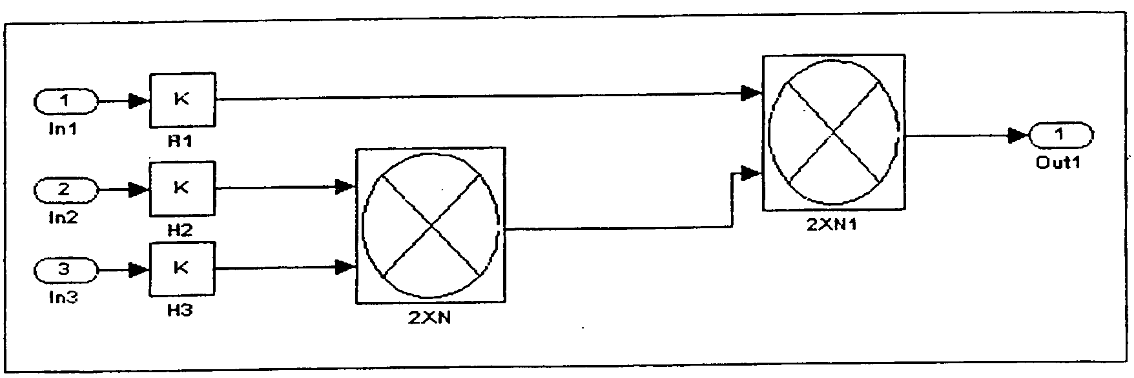



Fig. 25

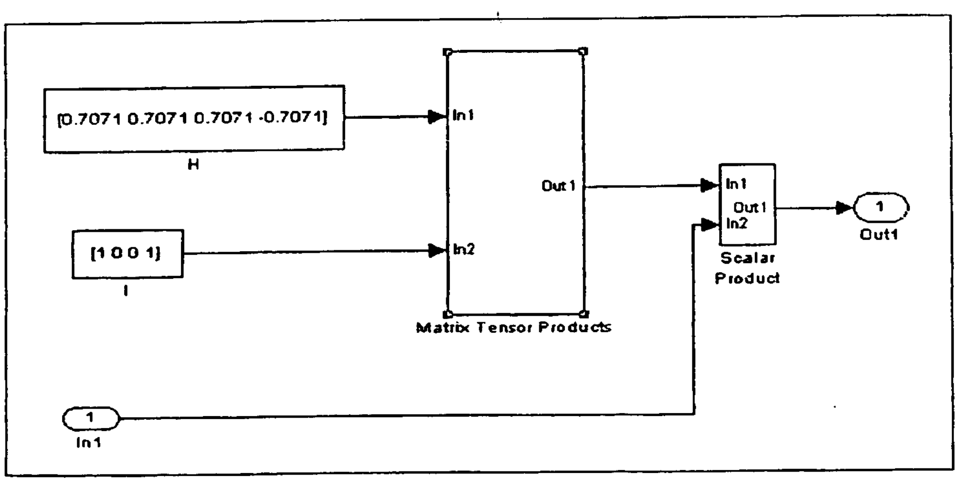



Fig. 26

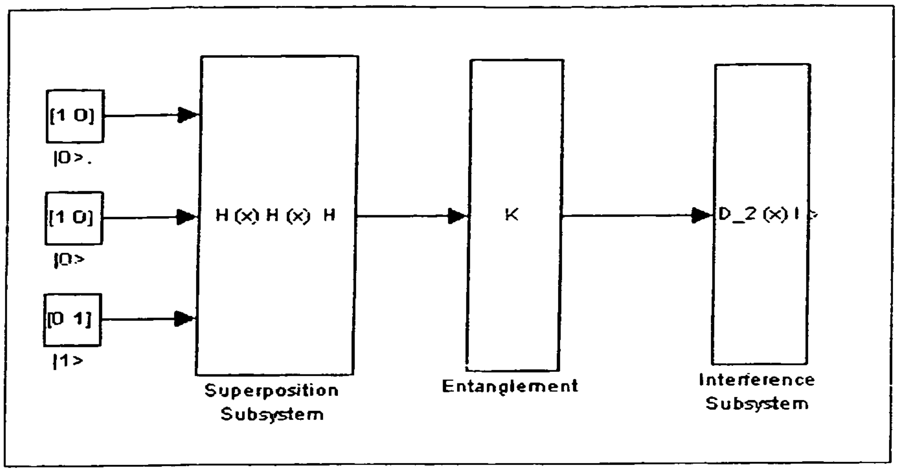



Fig. 27

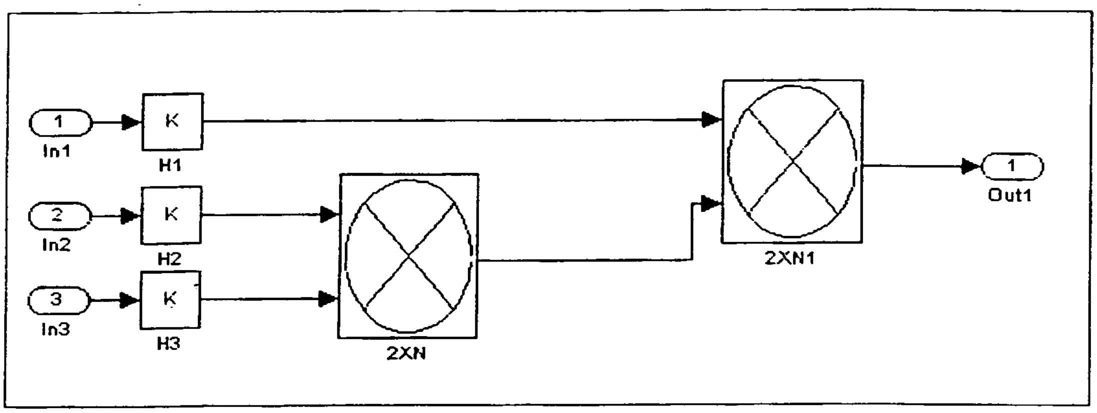



Fig. 28

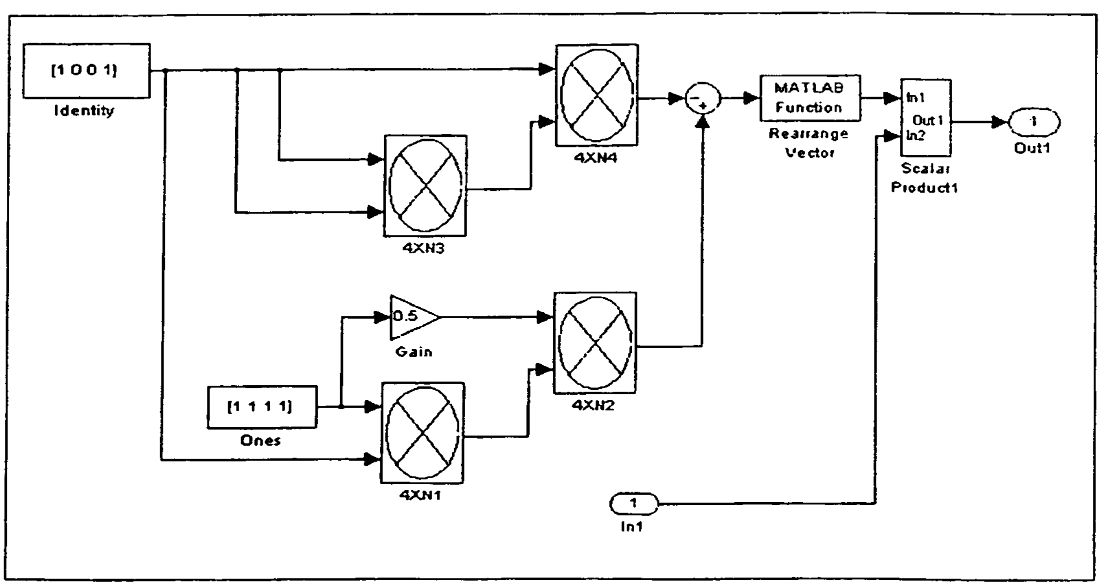



Fig. 29

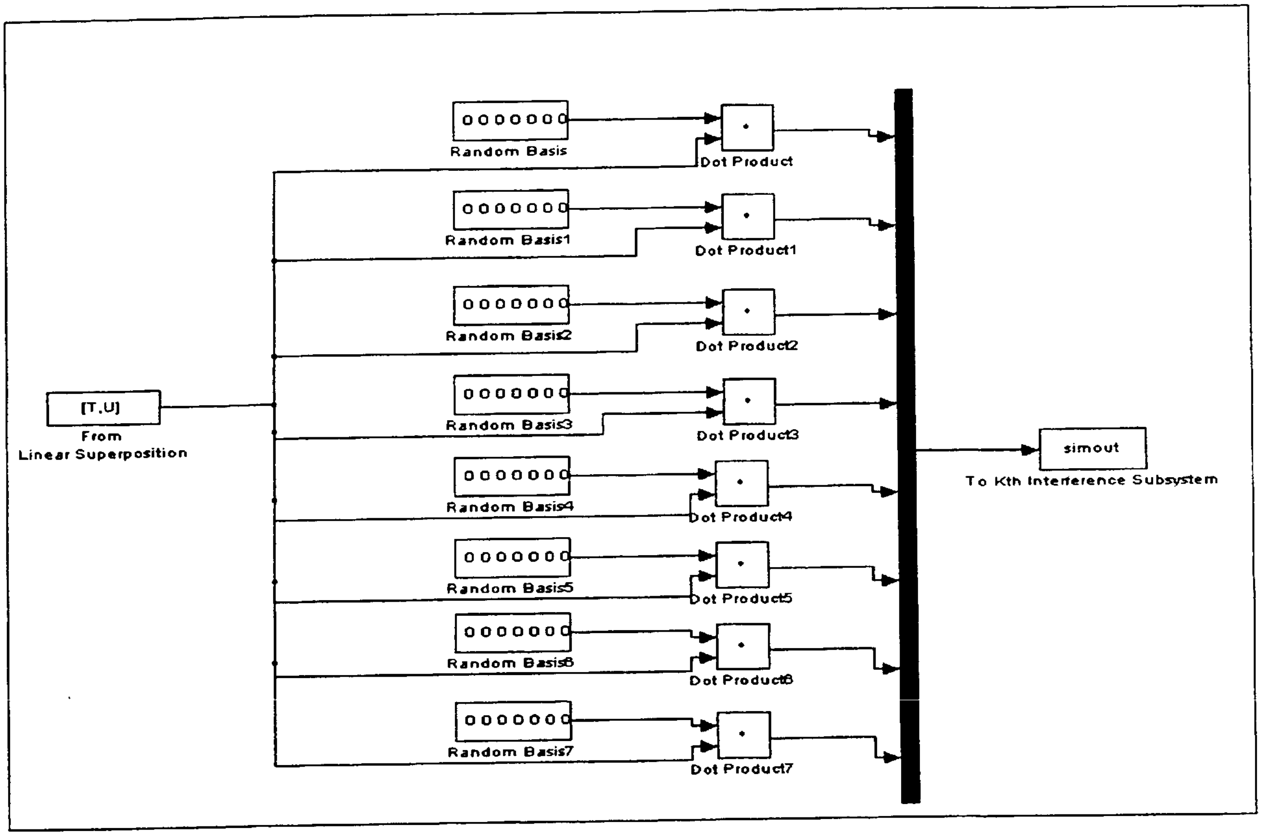

0000000
Random Basis
Dot Product
0000000
Random Basis1
Dot Product1
0000000
Random Basis2
Dot Product2
[T,U]
From
Linear Superposition
0000000
Random Basis3
Dot Product3
simout
To Kth Interference Subsystem
0000000
Random Basis4
Dot Product4
0000000
Random Basis5
Dot Product5
0000000
Random Basis6
Dot Product6
0000000
Random Basis7
Dot Product7

Fig. 30

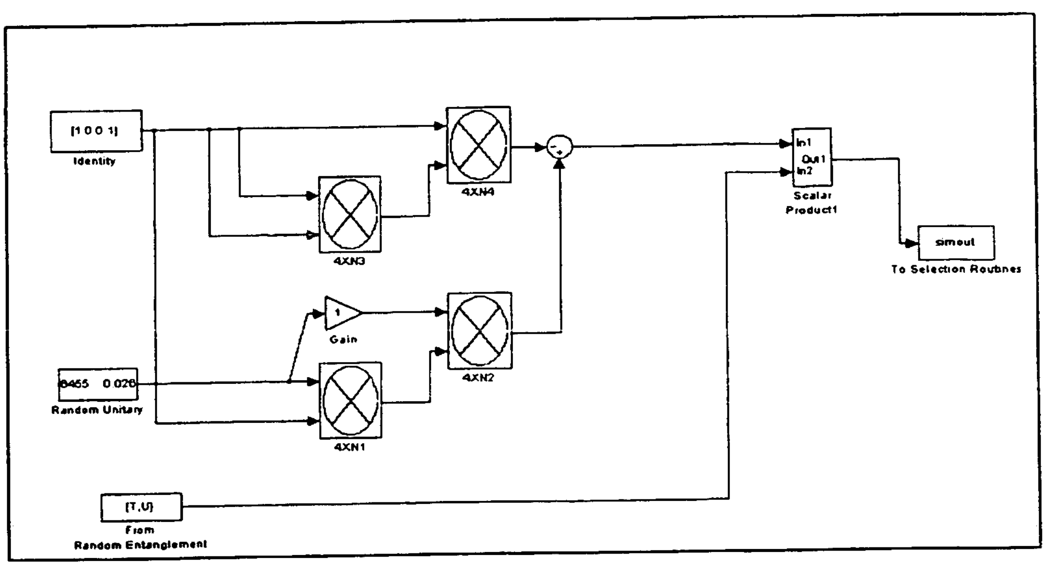

[1 0 0 1]
Identity
4XN4
4XN3
Scalar
Product1
simout
To Selection Routines
Gain
4XN2
Random Unitary
4XN1
[T,U]
From
Random Entanglement

Fig. 31

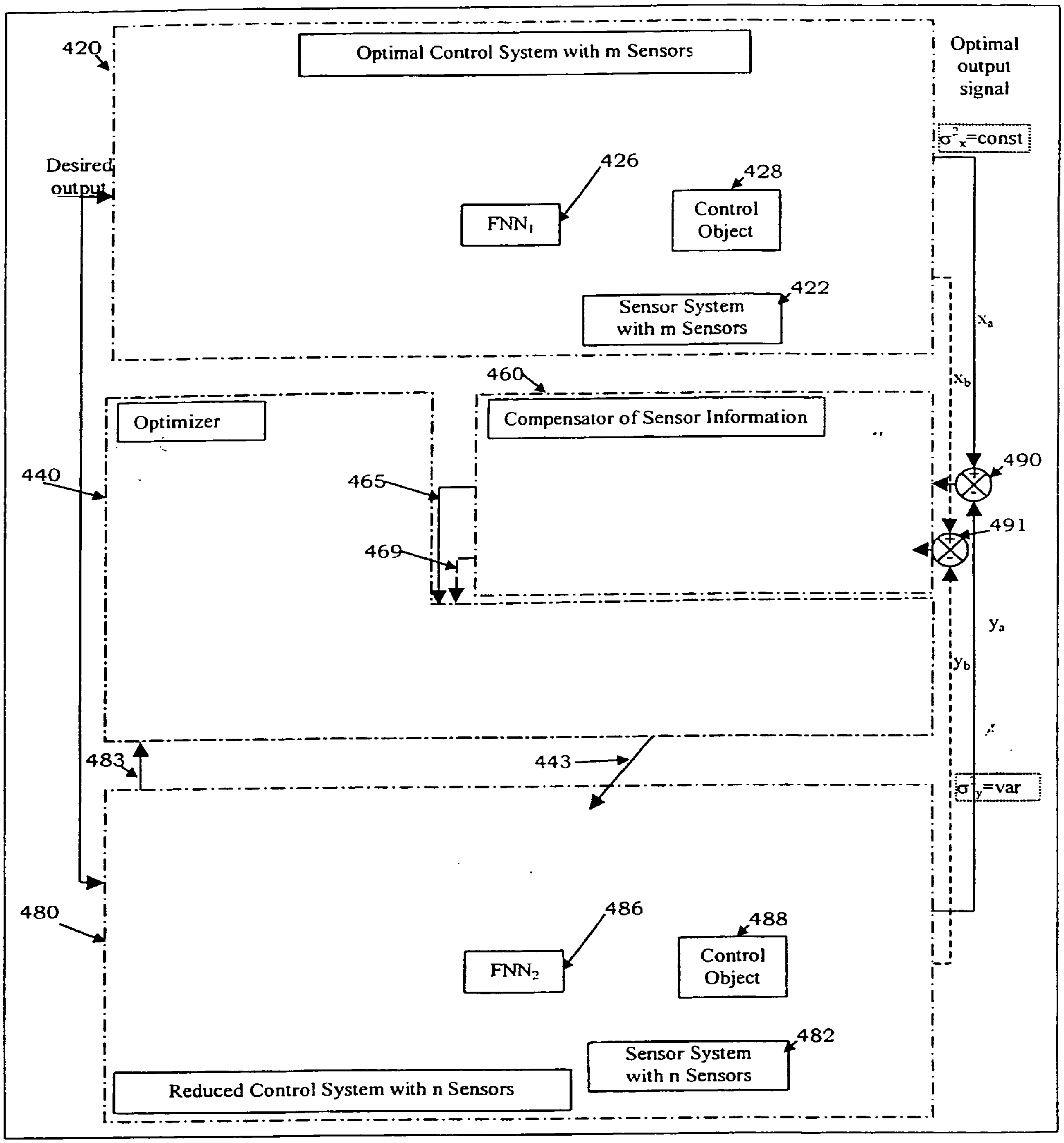
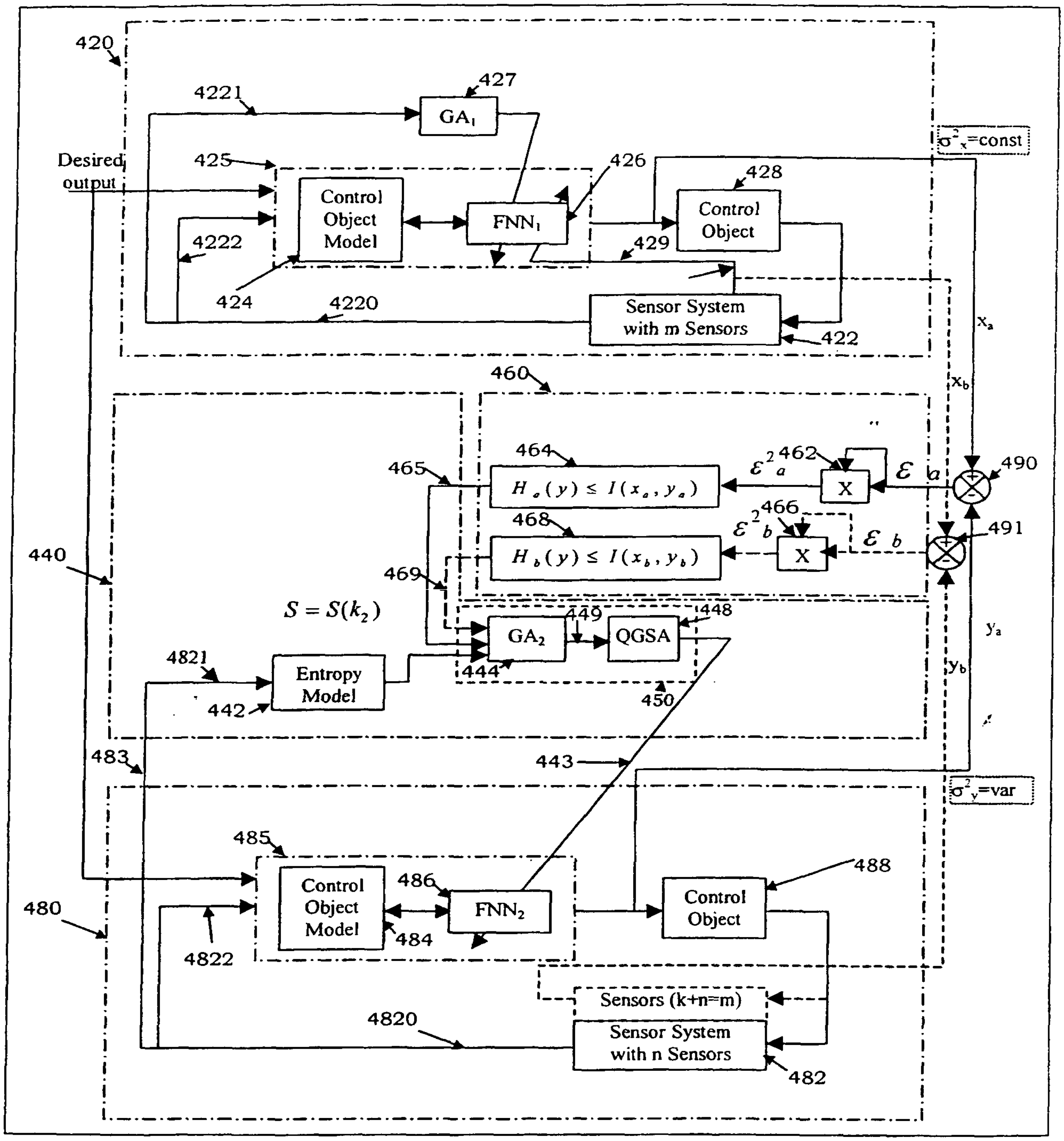

420
Optimal Control System with m Sensors
Optimal output signal
$\sigma^2_x$=const
Desired output
426
FNN$_1$
428
Control Object
422
Sensor System with m Sensors
$x_a$
460
$x_b$
Optimizer
Compensator of Sensor Information
440
465
490
491
469
$y_a$
$y_b$
483
443
$\sigma^2_y$=var
480
486
FNN$_2$
488
Control Object
482
Sensor System with n Sensors
Reduced Control System with n Sensors

Fig. 32



420
4221
427
GA₁
Desired output
425
426
428
Control Object Model
FNN₁
Control Object
4222
429
424
4220
Sensor System with m Sensors
422
σ²ₓ=const
xₐ
x_b
460
464
462
ε²ₐ
εₐ
X
490
465
Hₐ(y) ≤ I(xₐ, yₐ)
468
466
ε²_b
ε_b
X
491
H_b(y) ≤ I(x_b, y_b)
440
469
S = S(k₂)
449
448
GA₂
QGSA
yₐ
y_b
4821
Entropy Model
444
442
450
483
443
σ²_y=var
485
488
Control Object Model
486
FNN₂
Control Object
480
484
4822
Sensors (k+n=m)
4820
Sensor System with n Sensors
482


Fig. 33

**Table 4**

| | |
|---|---|
| 420 | Optimal Control System |
| 422 | Sensor System with m Sensors |
| 4220 | Information from $n=k_1+k_2$ Sensors |
| 4221 | Information from $k_2$ Sensors |
| 4222 | Information from $k_1$ Sensors |
| 424 | Control Object Model |
| 425 | Control Unit |
| 426 | Fuzzy Neural Network 1 |
| 427 | First Genetic Algorithm |
| 428 | Control Object |
| 429 | Teaching Signal |
| 440 | Optimizer |
| 442 | Thermodynamic Entropy Model |
| 443 | Teaching Signal |
| 444 | Second Genetic Algorithm |
| 448 | Quantum Genetic Search Algorithm |
| 450 | Global Optimizer |
| 460 | Compensator of Sensor Information |
| 462 | Multiplier |
| 464 | Information Calculator |
| 465 | $I(x_a, y_a)$ |
| 466 | Multiplier |
| 468 | Information Calculator |
| 469 | $I(x_b, y_b)$ |
| 480 | Reduced Control System |
| 482 | Sensor System with n=m-k Sensors |
| 4820 | Information from $n=k_1+k_2$ Sensors |
| 4821 | Information from $k_2$ Sensors |
| 4822 | Information from $k_1$ Sensors |
| 483 | Sensor Signal |
| 484 | Control Object Model |
| 485 | Reduced Control Unit |
| 486 | Fuzzy Neural Network 2 |
| 488 | Control Object |
| 490 | Subtractor |
| 491 | Subtractor |
| $x_a$ | Optimal Control Signal |
| $x_b$ | Output Sensor Signal |
| $y_a$ | Reduced Control Signal |
| $y_b$ | Output Sensor Signal |
| $\varepsilon_a$ | $x_a-y_a$ |
| $\varepsilon_b$ | $x_b-y_b$ |

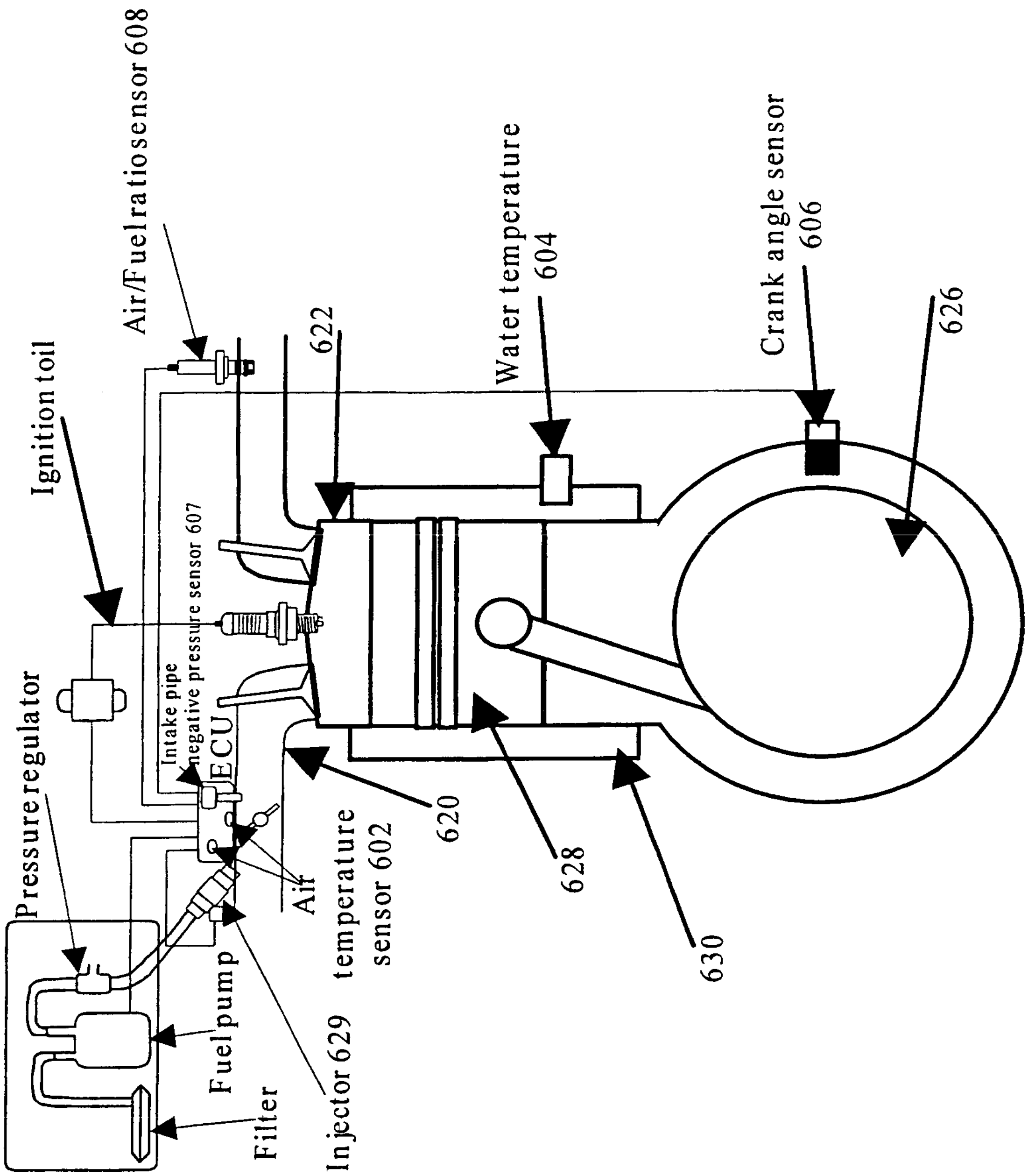

Air/Fuel ratio sensor 608
Ignition toil
Water temperature
604
Crank angle sensor
606
622
626
Pressure regulator
Intake pipe
negative pressure sensor 607
ECU
620
Air
temperature
sensor 602
628
630
Fuel pump
Injector 629
Filter

Fig. 34

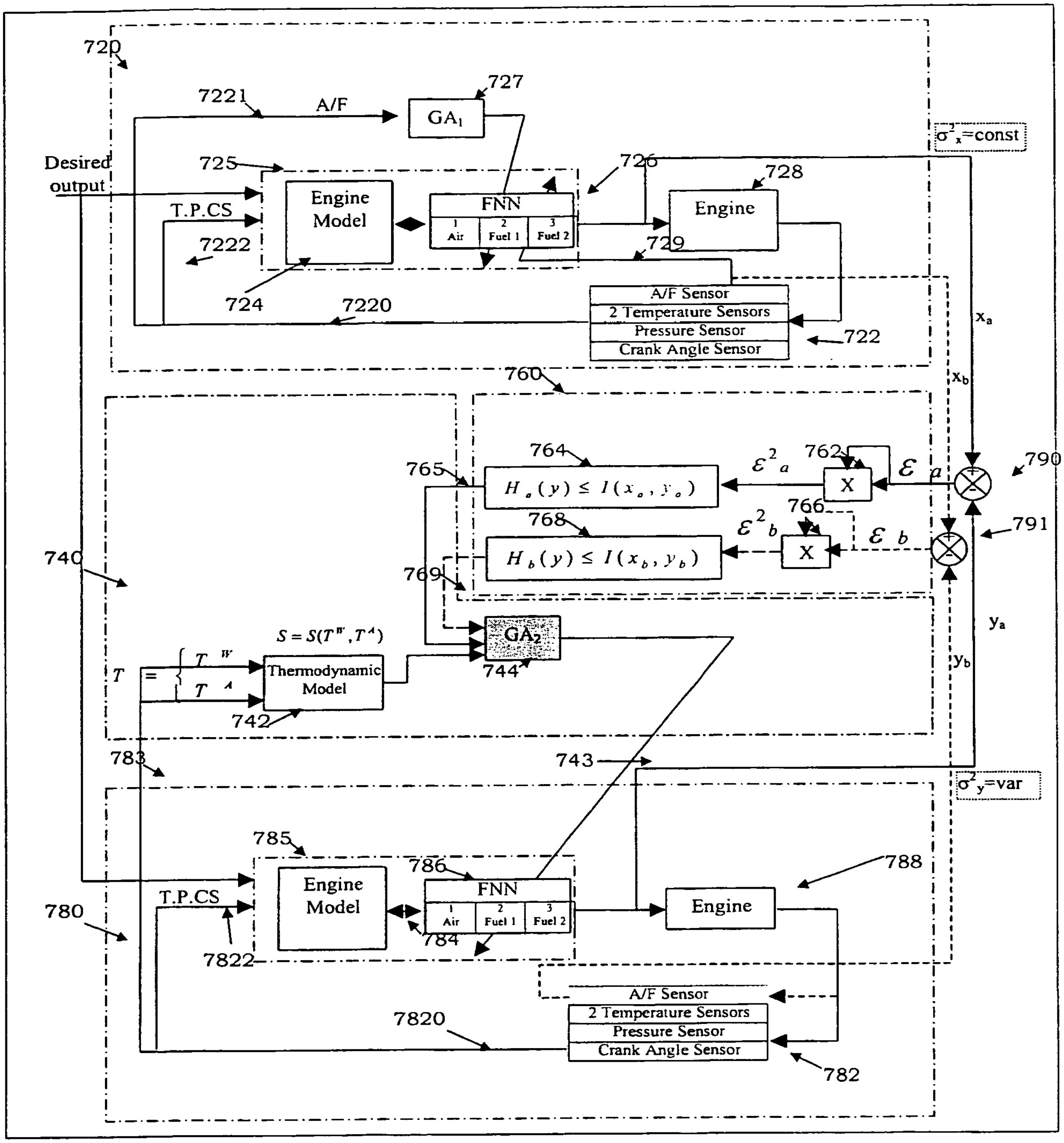

720
7221
A/F
GA1
727
Desired output
725
726
728
Engine Model
FNN
1 Air
2 Fuel 1
3 Fuel 2
Engine
T.P.CS
7222
729
724
7220
A/F Sensor
2 Temperature Sensors
Pressure Sensor
Crank Angle Sensor
722
σ²x=const
xa
xb
760
764
762
ε²a
εa
790
765
Ha(y) ≤ I(xa, ya)
766
768
ε²b
εb
791
Hb(y) ≤ I(xb, yb)
740
769
GA2
S = S(T^W, T^A)
T
Thermodynamic Model
744
742
ya
yb
783
743
σ²y=var
785
786
788
Engine Model
FNN
1 Air
2 Fuel 1
3 Fuel 2
Engine
780
T.P.CS
784
7822
7820
A/F Sensor
2 Temperature Sensors
Pressure Sensor
Crank Angle Sensor
782

Fig. 35

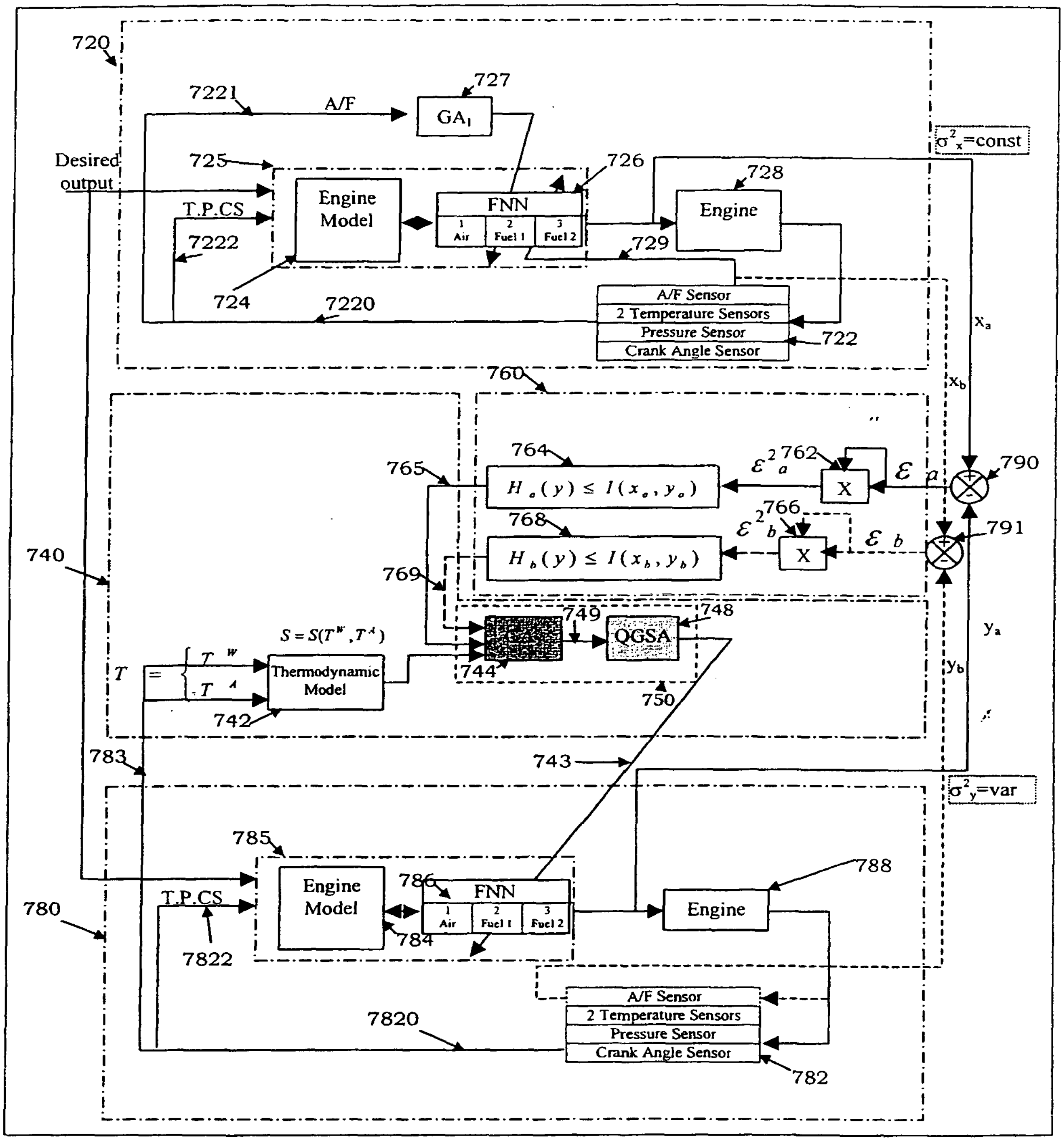

720
7221
A/F
GA1
727
Desired output
725
726
Engine Model
FNN
1 Air
2 Fuel 1
3 Fuel 2
728
Engine
T.P.CS
7222
729
724
7220
A/F Sensor
2 Temperature Sensors
Pressure Sensor
Crank Angle Sensor
722
σ²x=const
xa
xb
760
764
765
H a(y) ≤ I(xa, ya)
ε²a
762
X
ε a
790
768
H b(y) ≤ I(xb, yb)
766
ε²b
ε b
791
740
769
749
748
QGSA
S = S(T^W, T^A)
T
T W
T A
Thermodynamic Model
744
750
742
ya
yb
783
743
σ²y=var
785
788
786
Engine Model
FNN
1 Air
2 Fuel 1
3 Fuel 2
Engine
780
T.P.CS
784
7822
7820
A/F Sensor
2 Temperature Sensors
Pressure Sensor
Crank Angle Sensor
782

**Fig. 36**

# INTERNATIONAL SEARCH REPORT

International Application No
PCT/IT 00/00078

**A. CLASSIFICATION OF SUBJECT MATTER**
IPC 7 G05B13/02 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G05B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, WPI Data

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category * | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | US 5 971 579 A (KIM YOUNG-SANG) 26 October 1999 (1999-10-26) figure 2 --- | 1,8,20 |
| A | US 5 995 737 A (BONISSONE PIERO PATRONE ET AL) 30 November 1999 (1999-11-30) figure 6 --- | 1,8,20, 27 |
| A | WO 96 02025 A (SIEMENS AG ;HILLERMEIER CLAUS (DE); HOEHFELD MARKUS (DE); GEBERT R) 25 January 1996 (1996-01-25) page 4, line 11 -page 7, line 3 --- | 1,8,20 |
| A | US 5 796 077 A (JO SUNG-O) 18 August 1998 (1998-08-18) figure 1 --- | |

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 21 May 2001 | 30. 05. 2001 |

| Name and mailing address of the ISA | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,<br>Fax: (+31-70) 340-3016 | Katerbau, R |

3

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

| Category ° | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | HEY T: "QUANTUM COMPUTING: AN INTRODUCTION"<br>COMPUTING & CONTROL ENGINEERING JOURNAL,STEVENAGE,GB,<br>June 1999 (1999-06), pages 105-112, XP000937711<br>the whole document<br>--- | 13,16-19 |
| A | BOYER M ET AL: "Tight bounds on quantum searching"<br>FORTSCHRITTE DER PHYSIK, 1998, AKADEMIE VERLAG, GERMANY,<br>vol. 46, no. 4-5, pages 493-505, XP001001526<br>ISSN: 0015-8208<br>the whole document<br>--- | 13,16-19 |
| A | LLOYD S: "Quantum search without entanglement"<br>PHYSICAL REVIEW A (ATOMIC, MOLECULAR, AND OPTICAL PHYSICS), JAN. 2000, APS THROUGH AIP, USA,<br>vol. 61, no. 1, pages 010301/1-4, XP001003079<br>ISSN: 1050-2947<br>the whole document<br>--- | 14,15 |
| A | US 5 774 630 A (KIM JI-HYUN ET AL)<br>30 June 1998 (1998-06-30)<br>figure 1<br>--- | 27 |
| A | J.JONES ET AL: "IMPLEMENTATION OF A QUAMTUM SEARCH ALGORITHM ON A QUANTUM COMPUTER"<br>NATURE,<br>vol. 393, no. 6683,<br>28 May 1998 (1998-05-28), pages 344-346, XP000939342<br>UK<br>--- | |
| A | C.BARDEEN ET AL: "FEEDBACK QUANTUM CONTROL OF MOLECULAR ELECTRONIC POPULATION TRANSFER"<br>CHEMICAL PHYSICS LETTERS,<br>vol. 280, no. 1-2,<br>28 November 1997 (1997-11-28), pages 151-158, XP000937716<br>THE NETHRLANDS<br>---<br>-/-- | |

3

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

| Category ° | Citation of document, with indication,where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | A.OSEI ET AL: "USE OF QUANTUM INDETERMINACY IN OPTICAL PARALLEL GENETIC ALGORITHMS" PROCEEDINGS OF SPIE-THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING, vol. 2565, 10 July 1995 (1995-07-10), pages 192-197, XP000938185 USA | |

-----

3

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/IT 00/00078

**Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)**

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. [ ] Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. [ ] Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. [ ] Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

**Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)**

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. [X] As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. [ ] As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. [ ] As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. [ ] No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

**Remark on Protest**

[ ] The additional search fees were accompanied by the applicant's protest.

[X] No protest accompanied the payment of additional search fees.

**FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210**

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-12 20-26

   a METHOD FOR CONTROLLING A PROCESS USING NEURAL NETWORKS ,FUZZY LOGIC AND GENETIC ALGORTHMS

2. Claim : 13

   A METHOD FOR SEARCHING IN A DATABASE USING QUANTUM SEARCH ALGORITHM

3. Claim : 14

   A RANDOM ENTAGLEMENT GATE ACCORDING TO A QUANTUM SEARCH ALGORITHM

4. Claim : 15

   A RANDOM INTERFERENCE GATE ACCORDING TO A QUANTUM SEARCH ALGORTIHM

5. Claims: 16-19

   AN ACCELERATOR FOR QUANTUM ALGORITHM

6. Claims: 27-33

   A METHOD FOR TRAINING A FUZZY CONTROL UNIT

# INTERNATIONAL SEARCH REPORT

Int. ·tional Application No
PCT/IT 00/00078

| Patent document cited in search report | | Publication date | Patent family member(s) | | | Publication date |
|---|---|---|---|---|---|---|
| US 5971579 | A | 26-10-1999 | JP | 10031503 | A | 03-02-1998 |
| US 5995737 | A | 30-11-1999 | NONE | | | |
| WO 9602025 | A | 25-01-1996 | AT | 164460 | T | 15-04-1998 |
| | | | CN | 1151217 | A | 04-06-1997 |
| | | | DE | 59501727 | D | 30-04-1998 |
| | | | EP | 0770232 | A | 02-05-1997 |
| | | | ES | 2114747 | T | 01-06-1998 |
| | | | JP | 10502476 | T | 03-03-1998 |
| | | | RU | 2138840 | C | 27-09-1999 |
| | | | US | 5809488 | A | 15-09-1998 |
| US 5796077 | A | 18-08-1998 | JP | 9231080 | A | 05-09-1997 |
| US 5774630 | A | 30-06-1998 | CN | 1157866 | A,B | 27-08-1997 |
| | | | GB | 2307566 | A,B | 28-05-1997 |
| | | | JP | 9230909 | A | 05-09-1997 |
| | | | KR | 197710 | B | 15-06-1999 |